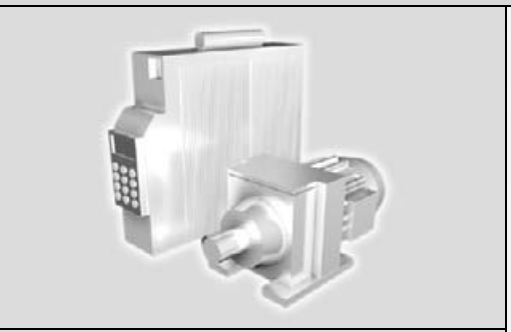
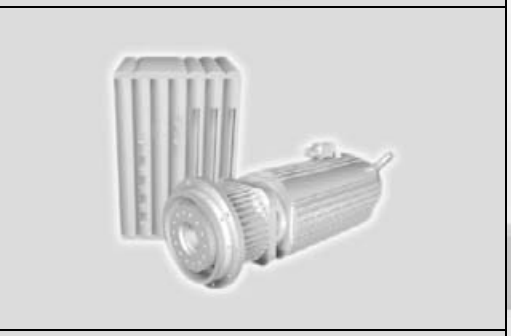
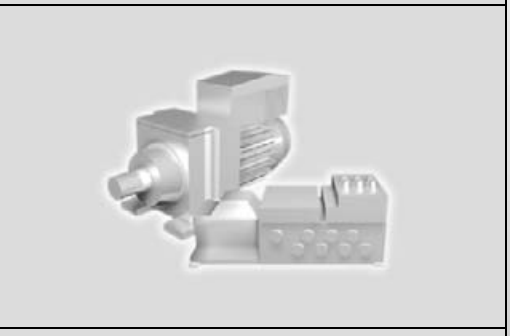
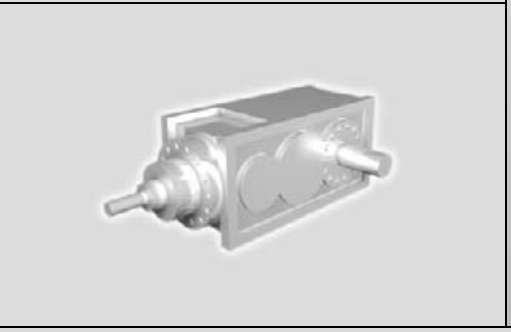
SEW EURODRIVE









MOVI-PLC®-Programmierung im PLC-Editor

EE330000

Ausgabe 12/2005

11362405 / DE

Systemhandbuch

SEW
EURODRIVE

1 Wichtige Hinweise 11

2 Systembeschreibung 12
2.1 Einführung in den PLC-Editor 12
2.2 Überblick zur Funktionalität des PLC-Editor 12
2.2.1 Wie ist ein Projekt strukturiert? 12
2.2.2 Wie erstelle ich ein Projekt? 12
2.2.3 Wie teste ich ein Projekt? 12
2.2.4 Debugging 13
2.3 Die Elemente des PLC-Editor 13
2.3.1 Bestandteile eines Projekts 13
2.4 Programmiersprachen 24
2.4.1 Unterstützte Programmiersprachen 24
2.4.2 Anweisungsliste AWL (IL) 24
2.4.3 Strukturierter Text ST (ST) 26
2.4.4 Ablaufsprache AS (SFC) 33
2.4.5 Funktionsplan FUP (FBD) 40
2.4.6 Freigrafischer Funktionsplaneditor CFC (CFC) 40
2.4.7 Kontaktplan KOP (LD) 41
2.5 Debugging 43
2.6 Die Norm IEC 61131-3 45

3 Programmierbeispiele 46
3.1 Voraussetzungen 46
3.2 Positionieren einer Motorachse 47
3.3 Drehzahlregelung einer Motorachse mit Geber 59
3.4 Tippbetrieb einer Motorachse ohne Geber 64

4 Komponenten des PLC-Editor 69
4.1 Das Hauptfenster 69
4.2 Projektoptionen 73
4.2.1 Menübefehl [Projekt] / [Optionen] ([Projects] / [Options]) 73
4.2.2 [Laden & Speichern] ([Load & Save]) 74
4.2.3 [Benutzerinformation] ([User information]) 75
4.2.4 [Editor] ([Editor]) 76
4.2.5 [Arbeitsbereich] ([Desktop]) 78
4.2.6 [Farben] ([Colors]) 80
4.2.7 [Verzeichnisse] ([Directories]) 81
4.2.8 [Logbuch] ([Log]) 83
4.2.9 [Übersetzungsoptionen] ([Build]) 85
4.2.10 [Kennworte] ([Passwords]) 88
4.2.11 [Sourcedownload] ([Source download]) 89
4.2.12 [Symbolkonfiguration] ([Symbol configuration]) 90
4.2.13 [Projektdatenbank] ([Database-connection]) 93
4.2.14 [Makros] ([Macros]) 97
4.3 Projekte verwalten 99
4.3.1 [Datei] / [Neu] ([File] / [New]) 99
4.3.2 [Datei] / [Neu aus Vorlage] ([File] / [New from template]) 99
4.3.3 [Datei] / [Öffnen] ([File] / [Open]) 99
4.3.4 [Datei] / [Schließen] ([File] / [Close]) 101
4.3.5 [Datei] / [Speichern] ([File] / [Save]) 101
4.3.6 [Datei] / [Speichern unter] ([File] / [Save as]) 102
4.3.7 [Datei] / [Archiv speichern/versenden] ([File] / [Save/Mail Archive]) 103
4.3.8 [Datei] / [Drucken] ([File] / [Print]) 106
4.3.9 [Datei] / [Einstellungen Dokumentation] ([File] / [Printer Setup]) 107
4.3.10 [Datei] / [Beenden] ([File] / [Exit]) 122

4.3.11 [Projekt] / [Übersetzen] ([Project] / [Build]) 122
4.3.12 [Projekt] / [Alles übersetzen] ([Project] / [Rebuild all]) 124
4.3.13 [Projekt] / [Alles bereinigen] ([Project] / [Clean all)] 124
4.3.14 [Projekt] / [Download-Information laden] ([Project] / [Load download information]) 124
4.3.15 [Projekt] / [In andere Sprache übersetzen] ([Project] / [Translate into other language]) 125
4.3.16 [Projekt] / [Dokumentieren] ([Project] / [Document]) 130
4.3.17 [Projekt] / [Exportieren] ([Project] / [Export]) 131
4.3.18 [Projekt] / [Importieren] ([Project] / [Import]) 132
4.3.19 [Projekt] / [Siemens Import] ([Project] / [Siemens Import]) 133
4.3.20 [Projekt] / [Vergleichen] ([Project] / [Compare]) 135
4.3.21 [Projekt] / [Kopieren] ([Project] / [Merge]) 139
4.3.22 [Projekt] / [Projektinformation] ([Project] / [Project Info]) 139
4.3.23 [Projekt] / [Global Suchen] ([Project] / [Global Search]) 141
4.3.24 [Projekt] / [Global Ersetzen] ([Project] / [Global replace]) 142
4.3.25 [Projekt] / [Überprüfen] ([Project] / [Check]) 143
4.3.26 Arbeitsgruppen 144
4.3.27 [Projekt] / [Projektdatenbank] ([Project] / [Project database]) 145
4.4 Objekte verwalten 156
4.4.1 Objekte 156
4.4.2 Ordner 156
4.4.3 [Knoten expandieren], [Knoten kollabieren] ([Expand Node], [Collapse Node]) 157
4.4.4 [Projekt] / [Objekt] / [Löschen] ([Project] / [Object] / [Delete]) 157
4.4.5 [Projekt] / [Objekt] / [Einfügen] ([Project] / [Object] / [Add]) 158
4.4.6 [Projekt] / [Objekt] / [Umbenennen] ([Project] / [Object]/ [Rename]) 159
4.4.7 [Projekt] / [Objekt] / [Konvertieren] ([Project] / [Object] / [Convert]) 160
4.4.8 [Projekt] / [Objekt] / [Kopieren] ([Project] / [Object] / [Copy]) 160
4.4.9 [Projekt] / [Objekt] / [Bearbeiten] ([Project] / [Object] / [Edit]) 161
4.4.10 [Projekt] / [Objekt]/[Eigenschaften] ([Project]/[Object]/[Properties]) 161
4.4.11 [Projekt] / [Aktion hinzufügen] ([Project] / [Add Action]) 163
4.4.12 [Projekt] / [Instanz öffnen] ([Project] / [View Instance]) 164
4.4.13 [Projekt] / [Querverweisliste ausgeben] ([Project] / [Show Cross Reference]) 165
4.4.14 [Projekt] / [Aufrufbaum ausgeben] ([Project] / [Show call tree]) 166
4.5 Allgemeine Editierfunktionen..... 167
4.5.1 Übersicht 167
4.5.2 [Bearbeiten] / [Rückgängig] ([Edit] / [Undo]) 167
4.5.3 [Bearbeiten] / [Wiederherstellen] ([Edit] / [Redo]) 168
4.5.4 [Bearbeiten] / [Ausschneiden] ([Edit] / [Cut]) 168
4.5.5 [Bearbeiten] / [Kopieren] ([Edit] / [Copy]) 169
4.5.6 [Bearbeiten] / [Einfügen] ([Edit] / [Paste]) 169
4.5.7 [Bearbeiten] / [Löschen] ([Edit] / [Delete]) 170
4.5.8 [Bearbeiten] / [Suchen] ([Edit] / [Find]) 170
4.5.9 [Bearbeiten] / [Weitersuchen] ([Edit] / [Find next]) 171
4.5.10 [Bearbeiten] / [Ersetzen] ([Edit] / [Replace]) 172
4.5.11 [Bearbeiten] / [Eingabehilfe] ([Edit] / [Input Assistant]) 173
4.5.12 [Bearbeiten] / [Variablen Deklaration] ([Edit] / [Auto declare]) 176
4.5.13 [Bearbeiten] / [Nächster Fehler] ([Edit] / [Next Error]) 176
4.5.14 [Bearbeiten] / [Vorheriger Fehler] ([Edit] / [Previous Error]) 177
4.5.15 [Bearbeiten] / [Makros] ([Edit] / [Macros]) 177
4.6 Allgemeine Online-Funktionen..... 178
4.6.1 Einleitung 178
4.6.2 [Online] / [Einloggen] ([Online] / [Login]) 178
4.6.3 [Online] / [Ausloggen] ([Online] / [Logout]) 180

4.6.4 [Online] / [Laden] ([Online] / [Download]) 180
4.6.5 [Online] / [Start] ([Online] / [Start]) 181
4.6.6 [Online] / [Stop] ([Online] / [Stop]) 181
4.6.7 [Online] / [Reset] ([Online] / [Reset]) 181
4.6.8 [Online] / [Reset (kalt)] ([Online] / [Reset (cold)]) 182
4.6.9 [Online] / [Reset (Ursprung)] ([Online] / [Reset (original)]) 182
4.6.10 [Online] / [Breakpoint an/aus] ([Online] / [Toggle Breakpoint]) 182
4.6.11 [Online] / [Breakpoint-Dialog] ([Online] / [Breakpoint Dialog]) 183
4.6.12 [Online] / [Einzelschritt über] ([Online] / [Step over]) 183
4.6.13 [Online] / [Einzelschritt in] ([Online] / [Step in]) 184
4.6.14 [Online] / [Einzelzyklus] ([Online] / [Single Cycle]) 184
4.6.15 [Online] / [Werte schreiben] ([Online] / [Write Values]) 184
4.6.16 [Online] / [Werte forcen] ([Online] / [Force Values]) 185
4.6.17 [Online] / [Forcen aufheben] ([Online] / [Release Force]) 186
4.6.18 [Online]/[Schreiben/Forcen-Dialog] ([Online]/[Write/Force-Dialog]) 187
4.6.19 [Online] / [Aufrufhierarchie] ([Online] / [Show Call Stack]) 189
4.6.20 [Online] / [Ablaufkontrolle] ([Online] / [Display Flow Control]) 189
4.6.21 [Online] / [Simulation] ([Online] / [Simulation Mode]) 190
4.6.22 [Online] / [Kommunikationsparameter] ([Online] / [Communication Parameters]) 190
4.6.23 [Online] / [Quellcode laden] ([Online] / [Sourcecode download]) ... 191
4.6.24 [Online] / [Bootprojekt erzeugen] ([Online] / [Create boot project]) 191
4.6.25 [Online]/ [Datei in Steuerung schreiben] ([Online] / [Write file to PLC]) 192
4.6.26 [Online]/[Datei aus Steuerung laden] ([Online] / [Read file from PLC]) 192
4.7 Fenster anordnen 193
4.7.1 Übersicht 193
4.7.2 [Fenster] / [Nebeneinander] ([Window] / [Tile vertical]) 193
4.7.3 [Fenster] / [Untereinander] ([Window] / [Tile horizontal]) 193
4.7.4 [Fenster] / [Überlappend] ([Window] / [Cascade]) 193
4.7.5 [Fenster] / [Symbole anordnen] ([Window] / [Arrange Symbols]) 193
4.7.6 [Fenster] / [Alle Schließen] ([Window] / [Close All]) 193
4.7.7 [Fenster] / [Meldungen] ([Window] / [Messages]) 193
4.7.8 [Fenster] / [Bibliotheksverwaltung] ([Window] / [Library Manager]) . 193
4.7.9 [Fenster] / [Logbuch] ([Window] / [Log]) 193
4.8 Die Online-Hilfe 194
4.8.1 [Hilfe] / [Inhalt] und [Suchen] ([Help] / [Contents] und [Search]) 194
4.8.2 Kontextsensitive Hilfe 195

5 PLC-Editor Programmierfenster 196
5.1 Editoren für die Programmiersprachen 196
5.1.1 Aufbau eines Editors 196
5.1.2 Kommentar 196
5.1.3 [Extras] / [Zoom] ([Extras] / [Zoom]) 197
5.1.4 [Extras] / [Instanz öffnen] 197
5.1.5 Intellisense-Funktion 197
5.2 Der Deklarationseditor 199
5.2.1 Allgemeines zum Deklarationseditor 199
5.2.2 Deklarationsteil 200
5.2.3 Eingabevariablen 201
5.2.4 Ausgabevariablen 201
5.2.5 Ein-/Ausgabevariablen 201
5.2.6 Lokale Variablen 202
5.2.7 Remanente Variablen 202
5.2.8 Konstanten (Typed Literals) 203
5.2.9 Externe Variablen 204
5.2.10 Schlüsselwörter 204
5.2.11 Variablendeklaration 204

5.2.12 AT-Deklaration ... 205
5.2.13 [Einfügen] / [Deklarationsschlüsselworte] ... 205
5.2.14 [Einfügen] / [Typen] ... 205
5.2.15 Syntaxcoloring ... 206
5.2.16 Kurzform-Modus ... 206
5.2.17 Automatisch deklarieren ... 207
5.2.18 Zeilennummer im Deklarationseditor ... 209
5.2.19 Deklaration als Tabelle ... 209
5.2.20 [Einfügen] / [Neue Deklaration] ([Insert] / [New Declaration]) ... 210
5.2.21 Deklaration im Online-Modus ... 210
5.3 Pragma-Anweisungen ... 212
5.3.1 Pragmas, Allgemeines ... 212
5.3.2 Pragmas für Initialisierung, Monitoring, Symbolerzeugung und für den Bit-Zugriff ... 212
5.3.3 Pragmas für Anzeige/Nicht-Anzeige von Deklarationsteilen im Bibliotheksverwalter ... 215
5.4 Editoren der textorientierten Programmiersprachen ... 217
5.4.1 Allgemeines zu Texteditoren ... 217
5.4.2 [Einfügen] / [Operator] in Texteditoren ... 217
5.4.3 [Einfügen] / [Operand] in Texteditoren ... 217
5.4.4 [Einfügen] / [Funktion] in Texteditoren ... 218
5.4.5 [Einfügen] / [Funktionsblock] in Texteditoren ... 218
5.4.6 Bausteinaufruf mit Ausgangsparametern in Texteditoren ... 218
5.4.7 Die Texteditoren im Online-Modus ... 219
5.4.8 [Extras]/[Monitoring Einstellungen] ([Extras]/[Monitoring Options]) ... 219
5.4.9 Breakpoint-Positionen im Texteditor ... 220
5.4.10 Was passiert an einem Breakpoint ... 221
5.4.11 Zeilennummern des Texteditors ... 221
5.4.12 Der Anweisungslisten-Editor AWL (IL) ... 221
5.4.13 Der Editor für Strukturierten Text ST ... 222
5.5 Die Editoren der grafisch orientierten Programmiersprachen ... 223
5.5.1 Die grafischen Editoren ... 223
5.5.2 Zoom ... 223
5.5.3 Netzwerk ... 224
5.5.4 Sprungmarken ... 224
5.5.5 Netzwerkkommentare, Umbrüche, [Extras] / [Optionen] ... 224
5.5.6 [Einfügen]/[Netzwerk (danach)] oder [Einfügen]/[Netzwerk (davor)] 227
5.5.7 Die Netzwerk-Editoren im Online-Modus ... 227
5.5.8 Der Funktionsplan-Editor FUP (FBD) ... 229
5.5.9 Der Kontaktplan-Editor KOP (LD) ... 237
5.5.10 Der Ablaufsprachen-Editor AS (SFC) ... 244
5.5.11 Der Freigrafische Funktionsplan-Editor CFC ... 256

6 PLC-Editor Ressourcen ... 276
6.1 Übersicht Ressourcen ... 276
6.2 Globale Variablen, Variablenkonfiguration, Dokumentvorlage ... 277
6.2.1 Objekte im Ordner [Globale Variablen] ... 277
6.2.2 Mehrere Variablenlisten ... 277
6.2.3 Globale Variablen ... 277
6.2.4 Variablenkonfiguration ... 280
6.2.5 Dokumentvorlage ... 281
6.3 Arbeitsbereich ... 283
6.4 Bibliotheksverwaltung ... 284
6.4.1 Bibliotheksverwalter ... 284
6.4.2 Bibliotheksverwalter nutzen ... 285
6.4.3 Standardbibliothek ... 285

6.4.4 Benutzerdefinierte Bibliotheken 285
6.4.5 [Einfügen] / [Weitere Bibliothek] 286
6.4.6 Bibliothek entfernen 286
6.4.7 [Extras] / [Eigenschaften] im Bibliotheksverwalter 286
6.5 Steuerungskonfiguration 287
6.5.1 Standardkonfiguration der Steuerung laden 287
6.5.2 Allgemeine Einstellungen in der Steuerungskonfiguration 287
6.5.3 Konfigurationsbaum 287
6.5.4 Übersicht der vorhandenen Unterelemente 290
6.5.5 Beispiel einer Steuerungskonfiguration 300
6.6 Taskkonfiguration 304
6.6.1 Überblick 304
6.6.2 [Einfügen] / [Task einfügen] oder [Einfügen] / [Task anhängen] 304
6.6.3 [Einfügen] / [Programmaufruf einfügen] oder [Einfügen] / [Programmaufruf anhängen] 306
6.6.4 System-Ereignisse 307
6.6.5 Welche Task wird bearbeitet? 308
6.6.6 [Extras] / [Debug Task festlegen] 309
6.6.7 [Extras] / [Task aus-/einschalten] 309
6.6.8 [Extras] / [Aufrufhierarchie] 309
6.7 Traceaufzeichnung 310
6.7.1 Überblick und Konfiguration 310
6.7.2 Eine Traceaufzeichnung durchführen 314
6.7.3 Betrachten der Traceaufzeichnung 314
6.7.4 [Extras] / [Tracewerte speichern] 317
6.7.5 [Extras] / [Externe Tracekonfiguration] 318
6.8 Watch- und Rezepturverwalter 320
6.8.1 Übersicht 320
6.8.2 Watch- und Rezepturverwalter im Offline-Modus 320
6.8.3 [Einfügen] / [Neue Watchliste] 321
6.8.4 [Extras] / [Watchliste umbenennen] 321
6.8.5 [Extras] / [Watchliste speichern] 322
6.8.6 [Extras] / [Watchliste laden] 322
6.8.7 Watch- und Rezepturverwalter im Online-Modus 322
6.8.8 [Extras] / [Monitoring aktiv] 323
6.8.9 [Extras] / [Rezeptur schreiben] 323
6.8.10 [Extras] / [Rezeptur lesen] 323
6.8.11 Werte forcen und schreiben im Watch Manager 324
6.9 Zielsystem Einstellungen 325
6.9.1 Übersicht 325
6.9.2 Dialog [Zielsystemeinstellungen] 325
6.9.3 Zielsystemeinstellungen DHP11B 326

7 Operatoren und Funktionen 331
7.1 Überblick 331
7.2 Arithmetische Operatoren 332
7.2.1 ADD 332
7.2.2 MUL 333
7.2.3 SUB 334
7.2.4 DIV 335
7.2.5 MOD 336
7.2.6 MOVE 337
7.2.7 INDEXOF 337
7.2.8 SIZEOF 338
7.3 Bitstring Operatoren 339
7.3.1 AND 339

7.3.2 OR ... 339
7.3.3 XOR ... 340
7.3.4 NOT ... 341
7.4 Bit-Shift Operatoren ... 342
7.4.1 SHL ... 342
7.4.2 SHR ... 343
7.4.3 ROL ... 344
7.4.4 ROR ... 345
7.5 Auswahloperatoren ... 346
7.5.1 Hinweise ... 346
7.5.2 SEL ... 346
7.5.3 MAX ... 347
7.5.4 MIN ... 348
7.5.5 LIMIT ... 349
7.5.6 MUX ... 350
7.6 Vergleichsoperatoren ... 351
7.6.1 GT ... 351
7.6.2 LT ... 352
7.6.3 GE ... 353
7.6.4 LE ... 354
7.6.5 EQ ... 355
7.6.6 NE ... 356
7.7 Adressoperatoren ... 357
7.7.1 ADR ... 357
7.7.2 ADRINST ... 357
7.7.3 BITADR ... 358
7.7.4 Inhaltsoperator ^ ... 358
7.8 Aufrufoperator ... 359
7.8.1 CAL ... 359
7.9 Typkonvertierungen ... 360
7.9.1 Hinweise ... 360
7.9.2 BOOL_TO-Konvertierungen ... 361
7.9.3 TO_BOOL-Konvertierungen ... 363
7.9.4 Konvertierungen zwischen ganzzahligen Zahlentypen ... 364
7.9.5 REAL_TO-/ LREAL_TO-Konvertierungen ... 365
7.9.6 TIME_TO- / TIME_OF_DAY-Konvertierungen ... 366
7.9.7 DATE_TO- / DT_TO-Konvertierungen ... 367
7.9.8 STRING_TO-Konvertierungen ... 368
7.9.9 TRUNC ... 369
7.10 Nummerische Operatoren ... 370
7.10.1 ABS ... 370
7.10.2 SQRT ... 371
7.10.3 LN ... 371
7.10.4 LOG ... 372
7.10.5 EXP ... 372
7.10.6 SIN ... 373
7.10.7 COS ... 373
7.10.8 TAN ... 374
7.10.9 ASIN ... 374
7.10.10 ACOS ... 375
7.10.11 ATAN ... 376
7.10.12 EXPT ... 376
7.11 Initialisierungsoperator ... 377
7.11.1 INI ... 377

8 Operanden 378
8.1 Konstanten 378
8.1.1 Konstanten vom Typ BOOL 378
8.1.2 Konstanten vom Typ TIME 378
8.1.3 Konstanten vom Typ DATE 379
8.1.4 Konstanten vom Typ TIME_OF_DAY 379
8.1.5 Konstanten vom Typ DATE_AND_TIME 379
8.1.6 Zahlenkonstanten 380
8.1.7 Konstanten vom Typ REAL und LREAL 380
8.1.8 Konstanten vom Typ STRING 381
8.1.9 Getypte Konstanten (Typed Literals) 381
8.2 Variablen 383
8.2.1 Variablen 383
8.2.2 Systemflags 383
8.2.3 Zugriff auf Variablen von Arrays, Strukturen und Bausteinen 384
8.2.4 Adressierung von Bits in Variablen 384
8.3 Adressen 386
8.3.1 Hinweis 386
8.3.2 Adressen 386
8.3.3 Merker 387
8.4 Funktionen 388

9 Datentypen 389
9.1 Überblick 389
9.2 Standard-Datentypen 389
9.2.1 Datentyp BOOL 389
9.2.2 Ganzzahlige Datentypen 389
9.2.3 Datentyp REAL und LREAL 390
9.2.4 Datentyp STRING 390
9.2.5 Die Zeitdatentypen 390
9.3 Definierte Datentypen 392
9.3.1 Datentyp Array 392
9.3.2 Pointer 394
9.3.3 Aufzählungstyp, Enumeration 395
9.3.4 Strukturen 396
9.3.5 Referenzen 397
9.3.6 Unterbereichstypen 397

10 Bibliotheken 401
10.1 Übersicht der Bibliotheken 401
10.2 Bibliothek standard.lib 402
10.2.1 String-Funktionen 402
10.2.2 Bistabile Funktionsblöcke 408
10.2.3 Flankenerkennung 411
10.2.4 Zähler 413
10.2.5 Timer 417
10.3 Bibliothek MPLCProcessdata.lib 421
10.3.1 Funktionsbaustein MC_ConnectAxis_PD 421
10.3.2 Funktionsbaustein MC_ReadParameter_PD 423
10.3.3 Funktionsbaustein MC_WriteParameter_PD 424

11 Anhang 426
11.1 Tastaturbedienung 426
11.1.1 Hinweis 426
11.1.2 Tastaturbefehl 427
11.2 Schlüsselwörter im PLC-Editor 430

11.3 PLC-Editor-Dateien 431
11.4 Warnungen und Übersetzungsfehler 433
11.4.1 Hinweis 433
11.4.2 Warnungen 433
11.4.3 Übersetzungsfehler 439

12 Index 476

1 Wichtige Hinweise

- **Dieses Systemhandbuch ersetzt nicht die ausführliche Betriebsanleitung!**
- **Die Steuerung MOVI-PLC® und die angesteuerten Antriebe nur durch Elektro-Fachpersonal unter Beachtung der gültigen Unfallverhütungsvorschriften, dem Systemhandbuch der Steuerung MOVI-PLC® und der Betriebsanleitung der Antriebe installieren und in Betrieb nehmen!**

Dokumentation

- Lesen Sie dieses Systemhandbuch sorgfältig durch, bevor Sie mit der Installation und Inbetriebnahme von Antrieben, die von der Steuerung MOVI-PLC® gesteuert werden, beginnen.
- Querverweise sind in diesem Systemhandbuch mit "→" gekennzeichnet. So bedeutet beispielsweise (→ Kap. X.X), dass Sie im Kapitel X.X dieses Systemhandbuches zusätzliche Informationen finden.
- Die Beachtung der Dokumentation ist die Voraussetzung für einen störungsfreien Betrieb und die Erfüllung eventueller Garantieansprüche.

Sicherheits- und Warnhinweise

Beachten Sie unbedingt die hier enthaltenen Sicherheits- und Warnhinweise!

Drohende Gefahr durch Strom.
Mögliche Folgen: Tod oder schwerste Verletzungen.

Drohende Gefahr.
Mögliche Folgen: Tod oder schwerste Verletzungen.

Gefährliche Situation.
Mögliche Folgen: Leichte oder geringfügige Verletzungen.

Schädliche Situation.
Mögliche Folgen: Beschädigung des Gerätes und der Umgebung.

Anwendungstipps und nützliche Informationen.

2 Systembeschreibung

2.1 Einführung in den PLC-Editor

Der PLC-Editor ist eine Entwicklungsumgebung für Steuerungen.

Der PLC-Editor ermöglicht dem SPS-Programmierer einen einfachen Einstieg in die mächtigen Sprachmittel der IEC. Die Benutzung der Editoren und der **Debugging**-Funktionen hat die ausgereiften Entwicklungsumgebungen höherer Programmiersprachen zum Vorbild (z. B. Visual C++).

2.2 Überblick zur Funktionalität des PLC-Editor

2.2.1 Wie ist ein Projekt strukturiert?

Ein Projekt, das alle Objekte des Steuerungsprogramms umfasst, wird in einer Datei abgelegt. Ein Projekt enthält verschiedene Arten von Objekten:

- Bausteine
- Datentypen-Definitionen
- Ressourcen

Der erste Baustein, den Sie in einem neuen Projekt **ohne** Taskkonfiguration anlegen, trägt automatisch den Namen PLC_PRG. Dort startet die Ausführung (entsprechend der main-Funktion in einem C-Programm), und von hier aus können Sie weitere Bausteine aufrufen:

- Programme
- Funktionsblöcke
- Funktionen

Wenn Sie eine Taskkonfiguration (Ressourcen) definiert haben, müssen Sie kein Programm mit Namen PLC_PRG angelegt werden. Näheres hierzu finden Sie auf Seite 304 (→ Kap. 6.6 Taskkonfiguration).

Im **Object Organizer** finden Sie alle Objekte Ihres Projekts aufgelistet.

2.2.2 Wie erstelle ich ein Projekt?

Zunächst müssen Sie die **Zielsystemeinstellungen** für Ihre Steuerung festlegen und gegebenenfalls angepassen.

Dann sollten Sie Ihre **Steuerung konfigurieren**, um die im Projekt verwendeten Ein- und Ausgangsadressen auf Korrektheit überprüfen zu können.

Anschließend legen Sie die notwendigen **Bausteine** an und programmieren in den gewünschten **Sprachen**.

Nach Abschluss der Programmierung **übersetzen** Sie das Projekt und beseitigen eventuell angezeigte Fehler.

Ein Beispiel für die Erstellung eines kleinen Projekts finden Sie auf Seite 46 (→ Kap. 3 Programmierbeispiele).

2.2.3 Wie teste ich ein Projekt?

Sind alle Fehler beseitigt, aktivieren Sie die **Simulation**, loggen sich in die simulierte Steuerung ein und "laden" Ihr Projekt in die Steuerung. Sie befinden sich nun im **Online-Modus**.

Öffnen Sie das Fenster mit Ihrer **Steuerungskonfiguration** und testen Sie Ihr Projekt

auf korrekten Ablauf. Belegen Sie hierzu manuell die Eingänge, und kontrollieren Sie, ob die Ausgänge wie gewünscht gesetzt werden. In den Bausteinen wird der Werteverlauf der lokalen Variablen angezeigt. Im **Watch- und Rezepturverwalter** konfigurieren Sie die Datensätze, deren Werte Sie betrachten wollen.

2.2.4 Debugging

Im Falle eines Programmierfehlers können Sie **Breakpoints** (Haltepunkte) setzen. Wenn die Ausführung in einem solchen Breakpoint stoppt, können Sie die Werte sämtlicher Projektvariablen zu diesem Zeitpunkt einsehen. Durch schrittweises Abarbeiten (**Einzelschritt**) können Sie Ihr Programm auf logische Korrektheit überprüfen.

Weitere Debugging-Funktionen:

- Programmvariablen und Ein/Ausgänge auf **bestimmte Werte setzen**
- Überprüfung der durchlaufenen Programmzeilen durch **Ablaufkontrolle**
- Chronologische Aufzeichnung der Vorgänge bzw. Benutzeraktionen und der internen Vorgänge während der Online-Sessions im **Logbuch**.
- Zyklusechte Aufzeichnung und Darstellung des Verlaufs von Variablen über einen längeren Zeitraum mit der **Traceaufzeichnung**. Diese Funktion muss in den Zielsystemeinstellungen aktiviert sein.

Wenn Sie das Projekt erstellt und getestet haben, laden Sie es in die Hardware und testen es dort. Es stehen Ihnen die gleichen **Debugging**-Funktionen wie bei der Simulation zur Verfügung.

2.3 Die Elemente des PLC-Editor

2.3.1 Bestandteile eines Projekts

Projekt

Ein Projekt beinhaltet alle Objekte eines Steuerungsprogramms. Ein Projekt wird in einer Datei mit dem Namen des Projekts gespeichert.

Zu einem Projekt gehören folgende Objekte:

- Bausteine
- Bibliotheken
- Datentypen
- Ressourcen

Baustein

Zu den Bausteinen gehören Funktionen, Funktionsblöcke und Programme. Diese können durch Aktionen ergänzt werden.

Jeder Baustein besteht aus einem Deklarationsteil und einem Code-Teil.

Der Code-Teil kann in einer der folgenden IEC-Programmiersprachen erstellt werden:

- AS (SFC)
- AWL (IL)
- CFC (CFC)
- FUP (FBD)
- KOP (LD)
- ST (ST)

Der PLC-Editor unterstützt alle IEC-Standardbausteine. Wenn Sie diese Bausteine in einem Projekt benutzen wollen, müssen Sie die Bibliothek `standard.lib` in Ihr Projekt einbinden.

Bausteine können andere Bausteine aufrufen. Rekursionen sind jedoch nicht erlaubt.

Funktion

Eine Funktion ist ein Baustein. Als Ergebnis der Ausführung wird genau ein Datum zurückgeliefert. Dieses Datum kann aus mehreren Elementen bestehen, wie z. B. Feldern oder Strukturen.

Der Aufruf einer Funktion kann in den textorientierten Programmiersprachen als ein Operator in Ausdrücken vorkommen.

Aufbau eines korrekten Deklarationkopfs einer Funktion:

```
FUNCTION Fct: INT
```

Eine Funktionsdeklaration beginnt mit dem Schlüsselwort `FUNCTION`, gefolgt vom Funktionsnamen (hier: `Fct`) mit Doppelpunkt. Nach dem Doppelpunkt müssen Sie den Funktionstyp angeben (hier: `INT`).

Der Funktion müssen Sie ein Ergebnis (Rückgabewert) zuweisen. Eine Funktion wird also wie eine Ausgabevariable benutzt.

Beispiel einer Funktion in der Sprache AWL

Deklarationsteil:

```
FUNCTION Fct: INT
VAR_INPUT
   PAR1 : INT;
   PAR2 : INT;
   PAR3 : INT;
END_VAR
```

Implementationsteil:

```
LD    PAR1
MUL   PAR2
DIV   PAR3
ST    Fct
```

Im Deklarationsteil werden drei Eingangsvariablen vom Typ *Integer* deklariert. Im Implementationsteil werden die ersten beiden Variablen miteinander multipliziert und anschließend durch die dritte Variable dividiert. Die Funktion liefert das Ergebnis der Rechenoperation zurück.

Hinweis

In ST können Sie einen Funktionsaufruf als Operand in Ausdrücken verwenden.

In der AS kann ein Funktionsaufruf nur innerhalb von Aktionen eines Schrittes oder in einer Transition erfolgen.

Aufruf einer Funktion

Beachten Sie beim Aufruf von Funktionen immer, dass diese über keine internen Zustände verfügen. Aufrufe einer Funktion mit denselben Argumenten (Eingabeparametern) liefern immer denselben Rückgabewert (Ausgabe). Deshalb dürfen Funktionen keine globalen Variablen und Adressen enthalten.

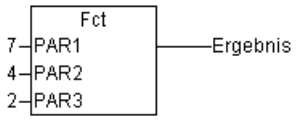
Aufruf einer Funktion in AWL

```
LD    7
Fct   2,4
ST    Ergebnis
```

Aufruf einer Funktion in ST

```
Ergebnis := Fct(7, 2, 4);
```

Aufruf einer Funktion in FUP



20239ADE

Achtung

Wenn eine lokale Variable in einer Funktion als `RETAIN` deklariert wird, hat dies keine Auswirkung. Die Variable wird nicht im Retain-Bereich gespeichert!

Hinweis

Wenn Sie in Ihrem Projekt eine Funktion mit Namen `CheckBounds` definieren, können Sie damit Bereichsüberschreitungen in Arrays automatisch überprüfen. Weitere Informationen dazu finden Sie auf Seite 389 (→ Kap. 9, Datentypen), .

Wenn Sie die Funktionen `CheckDivByte`, `CheckDivWord`, `CheckDivDWord` und `CheckDivReal` definieren, können Sie damit bei Verwendung des Operators `DIV` den Wert des Divisors überprüfen, beispielsweise um eine Division durch 0 zu verhindern. Weitere Informationen dazu finden Sie auf Seite 335 (→ Kap. 7.2.4, DIV), .

Wenn Sie die Funktionen `CheckRangeSigned` und `CheckRangeUnsigned` definieren, können Sie damit im Online-Modus automatisch Bereichsüberschreitungen bei Variablen abfangen, die mit Unterbereichstypen deklariert sind.

Die genannten Funktionsnamen sind aufgrund der hier beschriebenen Einsatzmöglichkeit reserviert.

Funktionsblock

Ein Funktionsblock – auch Funktionsbaustein genannt – ist ein Baustein, der bei der Ausführung einen oder mehrere Werte liefert.

Ein Funktionsblock liefert keinen Rückgabewert im Gegensatz zu einer Funktion.

Eine Funktionsblock-Deklaration beginnt mit dem Schlüsselwort `FUNCTION_BLOCK`.

Es können Kopien (Instanzen) eines Funktionsblocks geschaffen werden.

Der Aufruf eines Funktionsblocks erfolgt über eine Funktionsblockinstanz.

Beispiel eines Funktionsblocks in AWL

Deklarationsteil:

```
FUNCTION_BLOCK FUB

VAR_INPUT
   PAR1 : INT;
   PAR2 : INT;
END_VAR

VAR_OUTPUT
   MULERG : INT;
   VERGL : BOOL;
END_VAR
```

Implementationsteil:

```
LD    PAR1
MUL   PAR2
ST    MULERG

LD    PAR1
EQ    PAR2
ST    VERGL
```

Im Deklarationsteil werden zwei Eingabevariablen `PAR1` und `PAR2` und zwei Ausgabevariablen `MULERG` und `VERGL` deklariert.

Die Ausgabevariable `MULERG` ist das Produkt der beiden Eingabevariablen, `VERGL` ein Vergleich auf Gleichheit.

Instanzen von Funktionsblöcken

Es können Kopien (Instanzen), eines Funktionsblocks erzeugt werden.

Jede Instanz besitzt einen zugehörigen Bezeichner, den Instanznamen, und eine Datenstruktur, die ihre Eingabe- und Ausgabeparameter sowie ihre internen Variablen beinhaltet. Instanzen werden wie Variablen lokal oder global deklariert, indem der Name des Funktionsblocks als Typ eines Bezeichners angegeben wird.

Beispiel für eine Instanz mit Namen `INSTANZ` des Funktionsblocks `FUBO`:

```
INSTANZ : FUBO;
```

Funktionsblöcke werden stets über diese Instanzen aufgerufen.

Von außerhalb eines Funktionsblocks kann nur auf die Ein- und Ausgabeparameter einer Instanz zugegriffen werden, nicht auf dessen interne Variablen.

Nur die Deklarationsteile von Funktionsblöcken und Programmen können Instanzdeklarationen beinhalten. Instanzdeklarationen in Deklarationsteilen von Funktionen sind nicht zulässig.

Der Zugriff auf die Instanz eines Funktionsblocks ist auf den Baustein beschränkt, in

dem sie instanziert wurde, es sei denn, sie wurde global deklariert.

Der Instanzname kann als Eingabe einer Funktion oder eines Funktionsblocks benutzt werden.

Hinweis

Alle Werte bleiben zwischen zwei Ausführungen eines Funktionsblocks erhalten. Daher liefern mehrfache Aufrufe eines Funktionsblocks mit denselben Werten für die Eingabeparameter nicht immer dieselben Ausgabewerte!

Hinweis

Wenn der Funktionsblock mindestens eine Variable vom Typ `RETAIN` enthält, wird die gesamte Instanz im Retain-Bereich gespeichert.

Aufruf eines Funktionsblocks

Sie können die Eingabe- und Ausgabevariablen eines Funktionsblocks von einem anderen Baustein aus ansprechen, indem Sie eine Instanz des Funktionsblocks anlegen und über folgende Syntax die gewünschte Variable angeben:

`<Instanzname>.<Variablenname>`

Parameterwerte beim Aufruf zuweisen

Sie können den Eingabe- und/oder Ausgabevariablen beim Aufruf des Funktionsblocks einen Wert zuweisen.

In den textorientierten Programmiersprachen AWL und ST geben Sie nach dem Instanznamen die Parameterwerte in Klammern an.

Die Zuweisung geschieht bei Eingabevariablen mit `:=`, bei Ausgabevariablen mit =>.

Beispiel für ST – Aufruf eines Funktionsblocks über die Eingabehilfe

```
FBINST1(xx := WERT, yy => WERT);
```

`FBINST1` ist ein lokaler Funktionsblock (eine lokale Variable vom Typ eines Funktionsblocks), der die Eingabevariable `xx` und die Ausgabevariable `yy` enthält.

Hinweis

Wenn Sie eine Instanz unter Verwendung der Eingabehilfe (<F2>-Taste) mit der Option [Mit Argumenten] in das Implementationsfenster eines ST- oder AWL-Bausteins einfügen, wird die Instanz in dieser Syntax automatisch mit den entsprechenden Parametern dargestellt. Die Parameter müssen jedoch nicht zwingend belegt werden.

Ein-/Ausgabevariablen (E/A-Variablen) beim Aufruf

E/A-Variablen (`VAR_IN_OUT`) eines Funktionsblocks werden als Pointer übergeben. Den E/A-Variablen können Sie deshalb beim Aufruf keine Konstanten zuweisen. Ein Schreib- oder Lesezugriff auf diese Variablen von außerhalb des Funktionsblocks ist ebenfalls nicht möglich.

Beispiel in ST

```
VAR
   INST : FUBO;
   VAR1 : int;
END_VAR

VAR1 := 2;
INST(INOUT1 := VAR1);
```

Nicht zulässig ist: `INST(INOUT1:=2);` bzw. `INST.INOUT1:=2;`

Das Programm deklariert eine Instanz `INST` vom Typ des Funktionsblocks `FUBO` und eine Integer-Variable `VAR1`. Im Implementationsteil wird zuerst der Variablen `VAR1` der Wert "2" zugewiesen. In der letzten Zeile wird die Instanz `INST` des Funktionsblocks `FUNO` initialisiert. Dabei wird der E/A-Variablen `INOUT1` der Wert von `VAR1` zugewiesen.

Beispiele für den Aufruf des Funktionsblocks FUBO

Beispiel für den Aufruf des Funktionsblocks FUBO in AWL

Deklarationsteil:

```
FUNCTION_BLOCK FUBO

VAR_INPUT
   PAR1 : INT;
   PAR2 : INT;
END_VAR

VAR_OUTPUT
   MULERG : INT;
   VERGL  : BOOL;
END_VAR
```

Implementationsteil:

```
LD    PAR1
MUL   PAR2
ST    MULERG

LD    PAR1
EQ    PAR2
ST    VERGL
```

Die folgenden Programmcodes zeigen den Aufruf des Funktionsblocks `FUBO`. Es wird eine Instanz `INSTANZ` des Funktionsblocks deklariert. Beim Aufruf der Instanz werden den beiden Eingabevariablen Werte zugewiesen. Das Mulitplikationsergebnis dieser beiden Werte wird in der Variablen `ERG` abgelegt, das Ergebnis des Vergleichs der beiden Werte in `QUAD`.

Beispiele für Aufruf von FUBO in AWL

Deklarationsteil:

```
PROGRAM AWL_Aufruf
VAR
   QUAD    : BOOL;
   INSTANZ : FUBO;
   ERG     : INT:=0;
END_VAR
```

Implementationsteil:

```
CAL   INSTANZ(PAR1:=5, PAR2:=5)
LD    INSTANZ.VERGL
ST    QUAD

LD    INSTANZ.MULERG
ST    ERG
```

Beispiel für Aufruf von FUBO in ST

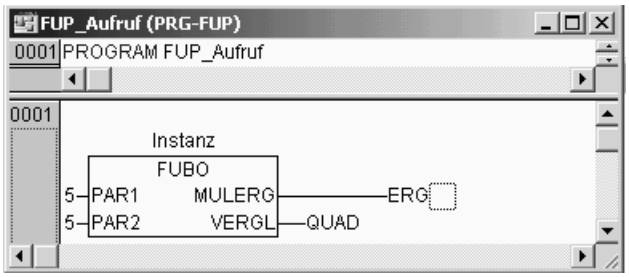
Der Deklarationsteil entspricht dem von AWL.

Implementationsteil:

```
PROGRAM ST_Aufruf

INSTANZ(PAE1:=5, PAR2:=5);   (bzw. INSTANZ;)
QUAD := INSTANZ.VERGL;
ERG := INSTANZ.MULERG;
```

Beispiel für einen Aufruf von FUBO in FUP

Der Deklarationsteil entspricht dem von AWL.



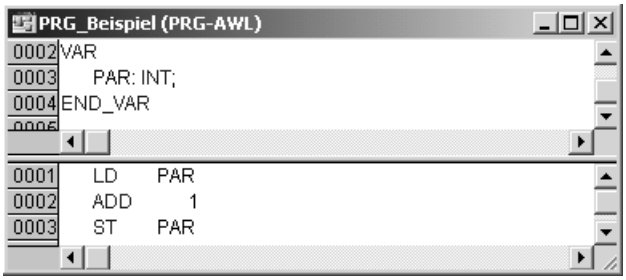
20240ADE

Beispiel für einen Aufruf von FUBO in AS

Der Aufruf von FUB0 ist in AS nicht möglich. In AS können Aufrufe von Funktionsblöcken nur in Schritten vorkommen.

Programm

Ein Programm ist ein Baustein, der bei der Ausführung einen oder mehrere Werte liefert. Programme sind global im gesamten Projekt bekannt. Alle Werte bleiben zwischen zwei Ausführungen des Programms erhalten.
Eine Programmdeklaration beginnt mit dem Schlüsselwort PROGRAM.



20241ADE

Programme können von Programmen und Funktionsblöcken aus aufgerufen werden. Ein Programmaufruf in einer Funktion ist nicht erlaubt.

Es gibt keine Instanzen von Programmen.

Wenn bei einem Programmaufruf aus einem Baustein Werte des Programms verändert werden, bleiben diese Veränderungen nach Verlassen des Programms erhalten. Dies gilt auch, wenn das Programm von einem anderen Baustein aus aufgerufen wird.

Dies ist anders beim Aufruf eines Funktionsblocks. Dort werden nur die Werte in der jeweiligen Instanz eines Funktionsblocks geändert. Diese Veränderungen spielen also auch nur eine Rolle, wenn dieselbe Instanz aufgerufen wird.

Parameterwerte beim Aufruf zuweisen

Sie können den Eingabe- und/oder Ausgabevariablen beim Aufruf eines Programms einen Wert zuweisen.

Hinweis

Wenn Sie ein Programm unter Verwendung der Eingabehilfe (<F2>-Taste) mit der Option [Mit Argumenten] im Implementationsfenster eines ST- oder AWL-Bausteins einfügen, wird das Programm in dieser Syntax automatisch mit den entsprechenden Parametern dargestellt. Die Parameter müssen jedoch nicht zwingend belegt werden.

Beispiele für Aufrufe von Programmen

In einem Programm `PRG_Beispiel2` sind die Eingabevariable `IN_VAR` und die Ausgabevariable `OUT_VAR` jeweils als Typ `INT` deklariert. Lokal deklariert ist die Variable `ERG`, ebenfalls als Typ `INT`.

Aufruf eines Programms in AWL

```
CAL   PRG_Beispiel2
LD    PRG_Beispiel2.OUT_VAR
ST    ERG
```

Eine andere Möglichkeit ist die unmittelbare Angabe der Parameter (Eingabehilfe mit der Option [Mit Argumenten]):

```
CAL   PRG_Beispiel2(IN_VAR:=33, OUT_VAR=>ERG)
```

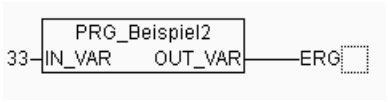
Aufruf eines Programms in ST

```
PRG_Beispiel2;
ERG := PRG_Beispiel2.OUT_VAR;
```

Eine andere Möglichkeit ist die unmittelbare Angabe der Parameter (Eingabehilfe mit der Option [Mit Argumenten]):

```
PRG_Beispiel2(IN_VAR:=33, OUT_VAR=>ERG);
```

Aufruf eines Programms in FUP



20242ADE

Beispiel für eine Aufrufsequenz von PLC_PRG

Das aufzurufende Programm ist PRG_Beispiel. Das Programm erhöht bei jedem Aufruf den Wert der Variablen par um "1".

```
LD   0
ST   PRG_Beispiel.PAR (* PAR wird mit 0 vorbesetzt *)
CAL  AWL_Aufruf (* ERG im AWL_Aufruf ergibt sich zu 1 *)
CAL  ST_Aufruf (* ERG im ST_Aufruf ergibt sich zu 2 *)
CAL  FUP_Aufruf (* ERG im FUP_Aufruf ergibt sich zu 3 *)
```

Das Hauptprogramm initialisiert die Variable PAR mit dem Wert "0". Anschließend wird das Programm PRG_Beispiel noch dreimal aufgerufen. Der in ERG gespeicherte Rückgabewert erhöht sich mit jedem Aufruf um "1". Dadurch wird deutlich, dass der Wert der Variablen PAR nach Verlassen des Programms PRG_Beispiel bis zum nächsten Aufruf erhalten bleibt.

PLC_PRG

Sie können die Projektabarbeitung über so genannte Tasks (Taskkonfiguration) zu steuern. Wenn jedoch keine Taskkonfiguration vorliegt, muss das Projekt den Baustein PLC_PRG enthalten. Der Baustein PLC_PRG vom Typ Programm wird automatisch erzeugt, wenn Sie in einem neu angelegten Projekt erstmalig mit [Projekt] / [Objekt einfügen] einen Baustein einfügen. Der Baustein PLC_PRG wird pro Steuerungszyklus genau einmal aufgerufen.

Liegt eine Taskkonfiguration vor, darf das Projekt keinen Baustein PLC_PRG enthalten, da dann die Ausführungsreihenfolge von der Taskzuordnung abhängt.

Achtung:

Löschen Sie den Baustein PLC_PRG nicht und benennen Sie ihn auch nicht um. PLC_PRG ist generell das Hauptprogramm in einem Single-Task-Programm.

Ressourcen

Ressourcen benötigen Sie zum Konfigurieren und Organisieren eines Projekts und zur Verfolgung von Variablenwerten:

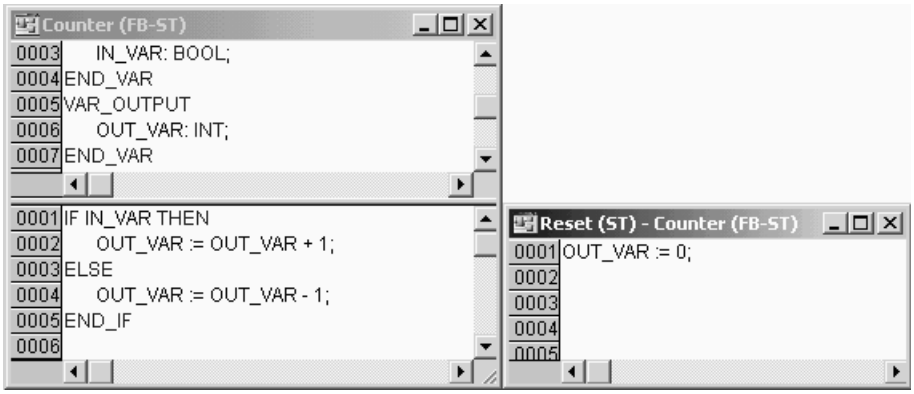
- **Globale Variablen** können Sie im gesamten Projekt bzw. Netzwerk verwenden.
- **Bibliotheken** können Sie über den Bibliotheksverwalter in das Projekt einbinden erden.
- **Logbuch** zum Aufzeichnen der Online-Aktivitäten
- **Steuerungskonfiguration** zum Konfigurieren der Hardware
- **Taskkonfiguration** zur Steuerung des Programms über Tasks
- **Watch- und Rezepturverwalter** zum Anzeigen und Vorbelegen von Variablenwerten
- **Zielsystemeinstellungen** zur Anwahl und gegebenenfalls Endkonfiguration des Zielsystems
- **Arbeitsbereich** mit einem Abbild der Projektoptionen
- **Traceaufzeichnung** zur grafischen Aufzeichnung von Variablenwerten

Aktion

Zu Funktionsblöcken und Programmen können Sie Aktionen definieren und hinzugefügen ([Projekt] / [Aktion hinzufügen] ([Project] / [Add Action])). Die Aktion stellt eine weitere Implementation dar, die in einer anderen Programmiersprache erstellt werden kann als die normale Implementation der Funktionsblock oder das Programm. Jede Aktion erhält einen Namen.

Eine Aktion arbeitet mit den Daten des Funktionsblocks bzw. Programms, zu dem sie gehört. Die Aktion verwendet die gleichen E/A-Variablen und lokalen Variablen wie die normale Implementation.

Beispiel für eine Aktion eines Funktionsblocks



20243ADE

In diesem Beispiel wird beim Aufruf des Funktionsblocks `COUNTER` die Ausgabevariable `out` um "1" erhöht bzw. erniedrigt in Abhängigkeit der Eingabevariablen `IN_VAR`. Bei Aufruf der Aktion `Reset` des Funktionsblocks wird die Ausgabevariable `OUT_VAR` auf "0" gesetzt. Es wird in beiden Fällen die gleiche Variable `OUT_VAR` beschrieben.

Aufruf einer Aktion

Eine Aktion rufen Sie auf mit `<Programmname>.<Aktionsname>` bzw. `<Instanzname>.<Aktionsname>`. Beachten Sie die Schreibweise in FUP (siehe Beispiel unten). Innerhalb des eigenen Bausteins rufen Sie eine Aktion in den textorientierten Programmiersprachen mit dem Aktionsnahmen auf, in den grafischorientierten Programmiersprachen den Funktionsblockaufruf ohne Instanzangabe.

Beispiele für Aufrufe einer Aktion aus einem anderen Baustein

Deklarationsteil für alle Beispiele:

```
PROGRAM PLC_PRG
VAR
   INSTANZ : Counter;
END_VAR
```

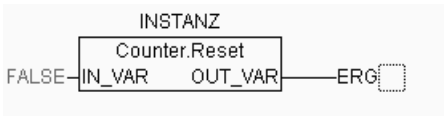
Aufruf der Aktion in AWL

```
CAL   INSTANZ.Reset(IN_VAR:=FALSE)
LD    INSTANZ.OUT_VAR
ST    ERG
```

Aufruf der Aktion in ST

```
INSTANZ.Reset(IN_VAR:=FALSE);
ERG := INSTANZ.OUT_VAR;
```

Aufruf der Aktion in FUP



20244ADE

Hinweis

Aktionen spielen bei Bausteinen in der Ablaufsprache AS eine besondere Rolle.

Bibliotheken

Sie können in ein Projekt eine Reihe von Bibliotheken einbinden. Bausteine, Datentypen und globale Variablen aus den Bibliotheken benutzen Sie genauso wie selbstdefinierte. Die Bibliotheken `standard.lib` und `util.lib` stehen Ihnen standardmäßig zur Verfügung.

Weitere Informationen hierzu finden Sie auf Seite 284 (→ Kap. 6.4 Bibliotheksverwaltung).

Datentypen

Ihnen steht eine Reihe von Standard-Datentypen zur Verfügung. Zusätzlich können Sie eigene Datentypen definieren. Sie können Strukturen, Aufzählungstypen und Referenzen anlegen.

Weitere Informationen hierzu finden Sie auf Seite 389 (→ Kap. 9 Datentypen).

2.4 Programmiersprachen

Dieses Kapitel gibt Ihnen einen kurze Einführung in die vom PLC-Editor unterstützten Programmiersprachen.

2.4.1 Unterstützte Programmiersprachen

Textorientierte Sprachen

- Anweisungsliste AWL (IL)
- Strukturierter Text ST (ST)

Grafisch orientierte Sprachen

- Ablaufsprache AS (SFC)
- Funktionsplan FUP (FBD)
- Kontaktplan KOP (LD)

Zusätzlich gibt es den Freigrafischen Funktionsplan CFC, der auf dem Funktionsplan basiert.

2.4.2 Anweisungsliste AWL (IL)

Eine Anweisungsliste AWL besteht aus einer Folge von Anweisungen. Jede Anweisung beginnt in einer neuen Zeile und beinhaltet einen Operator sowie – je nach Art der Operation – einen oder mehrere durch Kommas abgetrennte Operanden.

Identifikator-Marke

Vor einer Anweisung kann sich eine Identifikator-Marke befinden, gefolgt von einem Doppelpunkt (`:`). Er dient der Kennzeichnung der Anweisung und kann beispielsweise als Sprungziel verwendet werden.

Kommentare

Ein Kommentar beginnt mit der Zeichenfolge (`(*` und endet mit `*)`. Text zwischen diesen beiden Zeichenfolgen wird bei der Programmausführung ignoriert. Ein Kommentar muss das letzte Element in einer Zeile sein.

Whitespace

Leere Zeilen können zwischen Anweisungen eingefügt werden.

Operatoren

Es stehen folgende vordefinierte Operatoren zur Verfügung:

Operator	Modifikator	Bedeutung
LD	N	Setze aktuelles Ergebnis gleich dem Operanden.
ST	N	Speichere aktuelles Ergebnis an der Operandenstelle.
S		Setze den booleschen Operanden genau dann auf TRUE, wenn das aktuelle Ergebnis TRUE ist.
R		Setze den booleschen Operanden genau dann auf FALSE, wenn das aktuelle Ergebnis TRUE ist.
JMP	CN	Springe zur Marke.
CAL	CN	Rufe Programm oder Funktionsblock auf.
RET	CN	Verlasse den Baustein und kehre ggf. zum Aufrufer zurück.
)		Werte vorgestellte Operation aus.
Rechenoperationen		
ADD	(	Addition
SUB	(	Subtraktion
MUL	(	Multiplikation
DUV	(	Division
Vergleichsoperatoren		
EQ	(	=, gleich
NE	(	<>, ungleich
GT	(	>, größer als
LT	(	<, kleiner als
GE	(	>=, größer gleich
LE	(	<=, kleiner gleich
Bitweise Operationen		
AND	N, (	Bitweises AND
OR	N, (	Bitweises OR
XOR	N, (	Bitweises XOR (eXklusive OR)

Tabelle 1: Wichtige Operatoren

Eine Auflistung sämtlicher IEC-Operatoren finden Sie auf Seite 426 (→ Kap. 11, Anhang).

Modifikatoren

Es stehen folgende Modifikatoren zur Verfügung:

Modifikator	Operator	Bedeutung
C	JMP, CAL, RET	Die Anweisung wird nur ausgeführt, wenn das Ergebnis des vorhergehenden Ausdrucks TRUE ist.
N	JMPC, CALC, RETC	Die Anweisung wird nur ausgeführt, wenn das Ergebnis des vorhergehenden Ausdrucks FALSE ist.
	sonst	Negation des Operanden (nicht des Akkus).
(		Beginn eines Operanden, zum Erzeugen komplexerer mathematischer Ausdrücke.

Tabelle 2: Wichtige Modifikatoren

Programmbeispiel

```
LD   TRUE    (* Lade TRUE in den Akkumulator *)
ANDN BOOL1   (* Führe AND mit negiertem Wert der Variable BOOL1 aus *)
JMPC Marke   (* Wenn das Ergebnis TRUE ist, springe zur Marke "Marke" *)
LDN  BOOL2   (* Speichere den negierten Wert von *)
ST   ERG     (* BOOL2 in ERG *)

Marke:
LD   BOOL2   (* Speichere den Wert von BOOL2 ... *)
ST   ERG     (* ... in ERG *)
```

Klammern

Es ist in AWL auch möglich, Klammern nach einer Operation zu setzen. Als Operand wird dann der Wert der Klammer betrachtet.

Beispiel ohne Klammern

```
LD   2
MUL  2
ADD  3
ST   ERG
```

Die Variable ERG hat hier den Wert "7": 2 x 2 + 3 = 7.

Beispiel mit Klammern

```
LD   2
MUL  (2
ADD  3
)
ST   ERG
```

Die Variable ERG hat hier den Wert "10": 2 * (2 + 3) = 10.

2.4.3 Strukturierter Text ST (ST)

Überblick

Die Programmiersprache ST orientiert sich in ihrer Syntax an Hochsprachen wie Pascal. So gibt es neben den üblichen Operatoren z. B. auch einen Zuweisungsoperator und Schleifen wie IF und WHILE.

Beispiel

```
IF value < 7 THEN
   WHILE value < 8 DO
      value := value + 1;
   END_WHILE;
END_IF;
```

Ausdrücke

Ein Ausdruck ist ein Konstrukt, das einen Wert zurückliefert.

Ausdrücke sind zusammengesetzt aus Operatoren und Operanden.

Ein Operand kann sein:

- Konstante
- Variable
- Funktionsaufruf
- Literal
- weiterer Ausdruck

Das Ende eines Ausdrucks wird mit einem Semikolon (;) gekennzeichnet.

Auswertung von Ausdrücken

Die Auswertung eines Ausdrucks erfolgt durch Abarbeitung der Operatoren nach ihrer Priorität, also z. B. "Punktrechnung vor Strichrechnung". Der Operator mit der höchsten Priorität wird zuerst abgearbeitet.

Operatoren mit gleicher Prioriät werden von links nach rechts abgearbeitet.

Operatoren	Symbol	Priorität
Klammern	(Ausdruck)	höchste
Funktionsaufruf	Funktionsname (Parameterliste)	
Potenz	EXPT	
Negation, Komplementbildung	- NOT	
Multiplikation, Division, Modulo-Division	* / MOD	
Addition, Subtraktion	+ -	
Vergleiche	<, >, <=, >=	
Gleichheit, Ungleichheit	= <>	
Boolesches AND	AND	
Boolesches OR	OR	
Boolesches XOR	XOR	niedrigste

Tabelle 3: Operatoren in ST, sortiert nach absteigender Priorität

Anweisungen

Die folgende Tabelle zeigt die Anweisungen, die in ST zur Verfügung stehen, mit einem Beispiel.

Anweisung	Beispiel
Zuweisung (:=)	`A := B; CV := CV + 1; C := SIN(X)`
Aufruf eines Funktionsblocks und Benutzung der Funktionsblock-Ausgabe	`CMD_TMR(IN := %IX5, PT := 300);` `A := CMD_TMR.Q;`
RETURN	`RETURN;`

IF	`D := A * B;`
	`IF D < 0.0 THEN`
	`  C := A;`
	`ELSIF D = 0.0 THEN`
	`  C := B;`
	`ELSE`
	`  C := D;`
	`END_IF;`

CASE	`CASE INT1 OF`
	`  1: BOOL1 := TRUE;`
	`  2: BOOL2 := TRUE;`
	`ELSE`
	`  BOOL1 := FALSE;`
	`  BOOL2 := FALSE;`
	`END_CASE;`

FOR	`J := 101;`
	`FOR I:=1 TO 100 BY 2 DO`
	`  IF ARR[I] = 70 THEN`
	`  J := I;`
	`  EXIT;`
	`  END_IF;`
	`END_FOR;`

WHILE	`J := 1;`
	`WHILE J <= 100 AND ARR[J] <> 70 DO`
	`  J := J +2;`
	`END_WHILE;`

REPEAT	`J := -1;`
	`REPEAT`
	`  J := J + 2;`
	`UNTIL J = 101 OR ARR[J] = 70`
	`END_REPEAT;`

EXIT	`EXIT;`

Anweisung	Beispiel
Leere Anweisung	;

Zuweisungs-operator

Auf der linken Seite einer Zuweisung steht ein Operand (Variable, Adresse), dem der Wert des Ausdrucks auf der rechten Seite zugewiesen wird mit dem Zuweisungsoperator (:=).

Beispiel

```
VAR1 := VAR2 * 10;
```

Der Wert von VAR2 wird mit 10 multipliziert und das Ergebnis anschließend VAR1 zugewiesen.

Aufruf von Funktionsblöcken

Einen Funktionsblock rufen Sie auf, indem Sie den Namen der Instanz des Funktionsblocks aufrufen und in Klammern den Parametern die gewünschten Werte zuweisen.

Beispiel

```
CMD_TMR(IN := %IX5, PT := 300);
A := CMD_TMR.Q;
```

Zuerst wird wird ein Timer aufgerufen mit Zuweisungen für dessen Eingabevariablen IN und PT. Anschließend wird die Ausgabevariable Q an die Variable A zugewiesen. Auf die Ausgabevariable Q wird mit CMD_TMR.Q (Syntax: <Instanzname>.<Variablenname>) zugegriffen und der Wert anschließend der Variablen A zugewiesen.

RETURN-Anweisung

Die RETURN-Anweisung können Sie verwenden, um einen Baustein zu verlassen, beispielsweise in Abhängigkeit von einer Bedingung.

Beispiel

```
RETURN;
```

CASE-Anweisung

Mit der CASE-Anweisung können Sie mehrere bedingte Anweisungen für dieselbe Bedingungsvariable in ein Konstrukt zusammenfassen.

Syntax

```
CASE <VAR1> OF
   <Wert 1> :                  <Anweisung 1>
   <Wert 2> :                  <Anweisung 2>
   <Wert 3, Wert 4, Wert 5> :  <Anweisung 3>
   <Wert 6 .. Wert 10> :       <Anweisung 4>
   ...
   <Wert n> :                  <Anweisung m>
ELSE <ELSE-Anweisung>
END_CASE;
```

Eine CASE-Anweisung wird nach folgendem Schema abgearbeitet:

- Wenn die Variable in <VAR1> den Wert <Wert i> hat, dann wird die Anweisung <Anweisung i> ausgeführt.
- Hat <VAR1> keinen der angegebenen Werte, dann wird die <ELSE-Anweisung> ausgeführt.
- Wenn dieselbe Anweisung für mehrere Werte der Variablen ausgeführt werden soll, können Sie diese Werte durch Kommas voneinander getrennt hintereinander schreiben.

- Wenn dieselbe Anweisung für einen Wertebereich der Variablen ausgeführt werden soll, können Sie Anfangs- und Endwert durch zwei Punkte (`..`) getrennt hintereinander schreiben.

Beispiel

```
CASE INT1 OF
1, 5 :     BOOL1 := TRUE;
           BOOL3 := FALSE;
2 :        BOOL2 := FALSE;
           BOOL3 := TRUE;
10..20 :   BOOL1 := TRUE;
           BOOL3 := TRUE;
ELSE
           BOOL1 := NOT BOOL1;
           BOOL2 := BOOL1 OR BOOL2;
END_CASE;
```

IF-Anweisung

Mit der IF-Anweisung können Sie eine Bedingung prüfen und abhängig von dieser Bedingung die Anweisung ausführen.

Syntax

```
IF <Boolescher_Ausdruck1>
THEN
   <IF_Anweisungen>
{ELSIF  <Boolescher_Ausdruck2>
   THEN
      <ELSIF_Anweisungen1>
      .
      .
   ELSIF <Boolescher_Ausdruck n>
      THEN
         <ELSIF_Anweisungen n-1>
      ELSE
         <ELSE_Anweisungen>
}
END_IF;
```

Der Teil in geschweiften Klammern { } ist optional.

Wenn <Boolescher_Ausdruck1> TRUE ergibt, dann werden nur die <IF_Anweisungen> ausgeführt und keine der weiteren Anweisungen.

Anderenfalls werden die booleschen Ausdrücke, beginnend mit <Boolescher_Ausdruck2> der Reihe nach ausgewertet, bis einer der Ausdrücke TRUE ergibt. Dann werden nur die Anweisungen nach diesem booleschen Ausdruck und vor dem nächsten ELSE oder ELSIF ausgewertet.

Wenn keiner der booleschen Ausdrücke TRUE ergibt, dann werden ausschließlich die <ELSE-Anweisungen> ausgewertet.

Beispiel

```
IF TEMP < 17
   THEN heizung_an := TRUE;
   ELSE heizung_an := FALSE;
END_IF;
```

Die Heizung wird eingeschaltet, wenn die Temperatur unter 17 Grad absinkt, ansonsten bleibt sie aus.

FOR-Schleife

Mit der FOR-Schleife können Sie Vorgänge programmieren, die sich wiederholen.

Syntax

```
INT_VAR : INT;

FOR <INT_VAR> := <INIT_WERT> TO <END_WERT> {BY <Schrittgröße>}
DO
   <Anweisungen>
END_FOR;
```

Der Teil in geschweiften Klammern { } ist optional.

Die <Anweisungen> werden ausgeführt, solange der Zähler <INT_VAR> nicht größer als <END_WERT> ist. Dies wird vor der Ausführung der <Anweisungen> überprüft, sodass die <Anweisungen> niemals ausgeführt werden, wenn <INIT_WERT> größer als <END_WERT> ist.

Nachdem <Anweisungen> ausgeführt worden ist, wird <int_var> um <Schrittgröße> erhöht. Die Schrittgröße kann ein beliebiger Integerwert sein. Wenn die Schrittgröße nicht explizit angegeben ist, wird sie standardmäßig auf den Wert "1" gesetzt. Die Schleife muss also terminieren, da <INT_VAR> nur größer wird.

Hinweis:

<END_WERT> darf nicht gleich dem obersten Grenzwert des Wertebereichs des Zählers <INT_VAR> sein.

Beispiel: Wenn die Variable <INT_VAR> vom Typ SINT ist, darf <END_WERT> nicht 127 sein. Es würde sonst eine Endlosschleife entstehen, da <INT_VAR> niemals größer als 127 werden kann.

Beispiel

```
VAR1 : INT;
VAR1 := 1;

FOR ZAEHLER := 1 TO 5 BY 1
DO
   VAR1 := VAR1 * 2;
END_FOR;

ERG := VAR1;
```

Nach dem Durchlaufen der FOR-Schleife hat VAR1 den Wert "32".

WHILE-Schleife

Die WHILE-Schleife wird so lange ausgeführt, bis die Anfangsbedingung FALSE ergibt.

Syntax

```
WHILE <Boolescher Ausdruck> DO
   <Anweisungen>
END_WHILE;
```

Die <Anweisungen> werden so lange wiederholt ausgeführt wie <Boolescher_Ausdruck> TRUE ergibt. Wenn <Boolescher_Ausdruck> bereits bei der ersten Auswertung FALSE ist, dann werden die <Anweisungen> niemals ausgeführt. Wenn <Boolescher_Ausdruck> niemals den Wert FALSE annimmt, dann werden die <Anweisungen> endlos wiederholt, wodurch ein Laufzeitfehler entsteht.

Die WHILE-Schleife wird benutzt, wenn die Anzahl der Schleifendurchläufe zur Entwurfszeit nicht feststeht und die Schleifenbedingung vor dem ersten Durchlauf auch FALSE sein kann. Im letzteren Fall wird die Schleife – im Gegensatz zur REPEAT-Schleife – gar nicht abgearbeitet.

Hinweis:

Damit keine Endlosschleife entsteht, müssen Sie sicherstellen, dass die Schleifenbedingung im Anweisungsteil so verändert wird, dass <Booelscher Ausdruck> FALSE werden kann.

Beispiel

```
ZAEHLER, VAR1 : INT;
ZAEHLER := 5;
VAR1 := 1;

WHILE ZAEHLER <> 0 DO
   VAR1 := VAR1 * 2;
   ZAEHLER := ZAEHLER - 1;
END_WHILE;

ERG := VAR1;
```

Die WHILE-Schleife wird insgesamt fünfmal durchlaufen. Danach hat ZAEHLER den Wert "0", die Schleifenbedingung ist also FALSE. Nach dem Durchlaufen der WHILE-Schleife hat VAR1 den Wert "32".

REPEAT-Schleife

Die REPEAT-Schleife unterscheidet sich von der WHILE-Schleife dadurch, dass die Abbruchbedingung erst nach dem Ausführen der Schleife überprüft wird. Die Schleife wird also mindestens einmal durchlaufen.

Syntax

```
REPEAT
   <Anweisungen>
UNTIL <Boolescher Ausdruck>
END_REPEAT;
```

Die <Anweisungen> werden solange ausgeführt, bis <Boolescher Ausdruck> TRUE ergibt.

Wenn <Boolescher Ausdruck> bereits bei der ersten Auswertung TRUE ergibt, dann wird <Anweisungen> genau einmal ausgeführt. Wenn <Boolescher_Ausdruck> niemals den Wert TRUE annimmt, dann wird <Anweisungen> endlos wiederholt, wodurch ein Laufzeitfehler entsteht.

Die REPEAT-Schleife wird benutzt, wenn die Anzahl der Schleifendurchläufe zur Entwurfszeit unbekannt ist und die Schleife mindestens einmal durchlaufen werden soll.

Hinweis:

Damit keine Endlosschleife entsteht, müssen Sie sicherstellen, dass die Schleifenbedingung im Anweisungsteil so verändert wird, dass <Boolescher Ausdruck> FALSE werden kann.

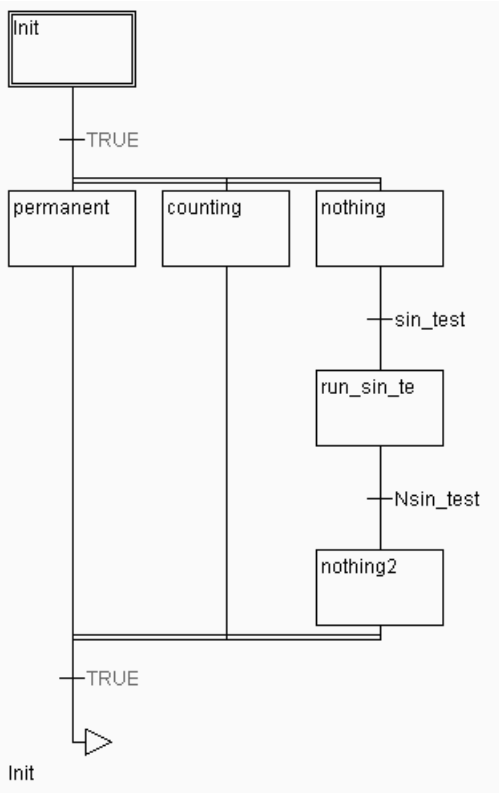
Beispiel

```
REPEAT
   VAR1 := VAR1 * 2;
   ZAEHLER := ZAEHLER - 1;
UNTIL ZAEHLER = 0
END_REPEAT;
```

EXIT-Anweisung Wenn die EXIT-Anweisung innerhalb einer FOR-, WHILE- oder REPEAT-Schleife vorkommt, wird diese Schleife beendet, ungeachtet der Abbruchbedingung.

2.4.4 Ablaufsprache AS (SFC)

Die Ablaufsprache AS ist eine grafisch orientierte Sprache, die es ermöglicht, die zeitliche Abfolge verschiedener Aktionen innerhalb eines Programms zu beschreiben. Dazu werden Schrittelemente verwendet, denen bestimmte Aktionen zugeordnet sind und deren Abfolge über Transitionselemente gesteuert wird.



20245ADE

Weitere Informationen zum Editor und zum Verhalten im Online-Modus:

- (→ Kap. 5.5.10 Der Ablaufsprachen-Editor AS (SFC)), Seite 244
- (→ Kap. 5.5.10, Die Ablaufsprache im Online-Modus), Seite 252

Schritt

Ein in der Ablaufsprache geschriebener Baustein besteht aus einer Folge von Schritten, die über gerichtete Verbindungen (Transitionen) miteinander verbunden sind.

Es gibt zwei Arten von Schritten:

- Ein einfacher Schritt besteht aus einer Aktion und einem Flag, das anzeigt, ob der Schritt aktiv ist. Wenn einem Schritt eine Aktion implementiert ist, erscheint ein kleines Dreieck in der rechten oberen Ecke des Schrittkästchens.
- Ein IEC-Schritt besteht aus einem Flag und einer oder mehreren zugewiesenen Aktionen oder booleschen Variablen. Die assoziierten Aktionen erscheinen rechts vom Schritt.

Aktion

Eine Aktion kann sein:

- eine Folge von Instruktionen in AWL oder in ST
- eine Menge von Netzwerken in FUP oder in KOP
- eine Ablaufstruktur in AS

Bei einfachen Schritten ist eine Aktion immer mit ihrem Schritt verbunden.

Sie können eine Aktion auf zwei Arten editieren:

- Doppelklicken Sie auf den Schritt, zu dem die Aktion gehört.
- Markieren Sie den Schritt, zu dem die Aktion gehört und und wählen Sie im Menü [Extras] / [Zoom Aktion/Transition] ([Extras] / [Zoom Action/Transition]) aus.

Pro Schritt sind eine Ein- und/oder Ausgangsaktion möglich.

Aktionen von IEC-Schritten hängen im **Object Organizer** direkt unter ihrem AS-Baustein und werden mit Doppelklick oder Drücken der <Eingabe>-Taste in ihren Editor geladen. Neue Aktionen können mit [Projekt] / [Aktion hinzufügen] ([Project] / [Add Action]) erzeugt werden. Einem IEC-Schritt können maximal neun Aktionen hinzugefügt werden.

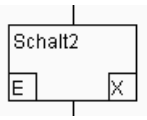
Eingangs- und Ausgangsaktion

Einem Schritt kann zusätzlich zur Schritt-Aktion eine Eingangsaktion und eine Ausgangsaktion hinzugefügt werden. Eingangs- und Ausgangsaktion werden jeweils nur einmal ausgeführt: Die Eingangsaktion gleich nachdem der Schritt aktiv geworden ist, die Ausgangsaktion als Letztes bevor der Schritt deaktiviert wird.

Ein Schritt mit Eingangsaktion wird durch ein "E" in der linken unteren Ecke gekennzeichnet, einer mit Ausgangsaktion durch ein "X" in der rechten unteren Ecke.

Die Ein- und Ausgangsaktion kann in einer beliebigen Sprache implementiert werden. Um eine Ein- bzw. Ausgangsaktion zu editieren, doppelklicken Sie auf die entsprechende Ecke im Schritt.

Beispiel für einfachen Schritt mit Eingangs- und Ausgangsaktion



20246ADE

Transition und Transitionsbedingung

Zwischen den Schritten liegen sogenannte Transitionen.

Eine Transitionsbedingung kann den Wert TRUE oder FALSE haben. Somit kann sie aus einer booleschen Variablen, einer booleschen Adresse oder einer booleschen Konstante bestehen.

Sie kann auch eine Folge von Instruktionen mit einem booleschen Ergebnis in ST-Syntax (z. B. `(i <= 100) AND b`) oder einer beliebigen Sprache enthalten (Menübefehl [Extras] / [Zoom Aktion/Transition] ([Extras] / [Zoom Action/Transition])).

HINWEIS

Eine Transition darf keine Programme, Funktionsblöcke oder Zuweisungen enthalten!

Im AS-Editor kann eine Transitionsbedingung direkt an die Transitionsmarke geschrieben werden oder es kann ein eigenes Editorfenster dafür geöffnet werden (Menübefehl [Extras] / [Zoom Aktion/Transition]). Die im Editor vorliegende Bedingung hat Vorrang!

Zur Analyse von Transitionsausdrücken kann das Flag `SFCErrorAnalyzationTable` definiert werden.

HINWEIS

Neben Transitionen kann auch der Tip-Modus benutzt werden, um zum nächsten Schritt weiterzuschalten. Informationen zu den beiden Flag-Variablen `SFCtip` und `SFCTip-mode` finden Sie im (→ Abschnitt. Flags), Seite 38.

Aktiver Schritt

Nach dem Aufruf des AS-Bausteins wird zunächst die zum Initialschritt (doppelt umrandet) gehörende Aktion ausgeführt. Ein Schritt, dessen Aktion ausgeführt wird, gilt als "aktiv". Im Online-Modus werden aktive Schritte blau dargestellt.

Pro Zyklus werden zunächst alle Aktionen ausgeführt, die zu aktiven Schritten gehören. Danach werden die Schritte auf "aktiv" gesetzt, die den aktiven Schritten folgen, wenn die Transitionsbedingungen für diese nachfolgenden Schritte TRUE sind. Die nun aktiven Schritte werden im nächsten Zyklus durchgeführt.

HINWEIS

Wenn der aktive Schritt eine Ausgangsaktion enthält, wird diese erst im nächsten Zyklus ausgeführt, vorausgesetzt, die darauffolgende Transition ist TRUE.

IEC-Schritt

Neben den einfachen Schritten stehen die normkonformen IEC-Schritte in AS zur Verfügung.

Um IEC-Schritte verwenden zu können, müssen Sie die SFC-Bibliothek `lecsfc.lib` in das Projekt einbinden.

Einem IEC-Schritt können Sie maximal neun Aktionen zugweisen werden. IEC-Aktionen sind im Gegensatz zu den Eingangs- und Ausgangsaktionen eines einfachen Schritts nicht fest dem IEC-Schritt zugeordnet. Deshalb können Sie IEC-Aktionen innerhalb eines Bausteins mehrfach verwenden. Über den Menübefehl [Extras] / [Aktion assoziieren] ([Extras] / [Associate Action]) ordnen Sie einem IEC-Schritt eine IEC-Aktion zu.

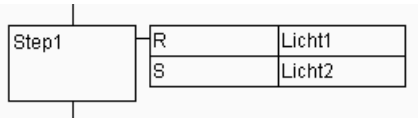
Sie können Schritten neben Aktionen auch boolesche Variablen zuweisen.

Über so genannte Qualifier (Bestimmungszeichen) wird die Aktivierung bzw. Deaktivierung der Aktionen und booleschen Variablen gesteuert. Dabei sind zeitliche Verzögerungen möglich, da eine Aktion immer noch aktiv sein kann, wenn bereits der nächste Schritt abgearbeitet wird. Solche Nebenläufigkeiten können z. B. durch den Qualifier S (Set) erreicht werden.

Eine assoziierte boolesche Variable wird bei jedem neuen Aufruf des AS-Bausteins gesetzt oder zurückgesetzt. Es wird dabei der Variablen jeweils der Wert TRUE oder FALSE zugewiesen.

Die assoziierten Aktionen werden rechts vom IEC-Schritt in einem zweigeteilten Kästchen dargestellt. Das linke Feld enthält den Qualifier, eventuell mit Zeitkonstanten. Das rechte Feld enthält den Namen der Aktion bzw. der Variablen.

Beispiel für einen IEC-Schritt mit zwei Aktionen



20247ADE

Zur leichteren Verfolgung der Vorgänge werden im Online-Modus alle aktiven Aktionen – analog den aktiven Schritten – blau dargestellt. Nach jedem Zyklus wird überprüft, welche Aktionen aktiv sind.

Beachten Sie hierzu auch die Einschränkung, wenn Sie von Zeit-Qualifier bei Aktionen verwenden, die in einem Zyklus mehrfach verwendet werden.

HINWEIS

Wenn eine Aktion deaktiviert wird, wird sie **noch einmal** ausgeführt. Jede Aktion wird also mindestens zweimal ausgeführt. Das gilt auch für eine Aktion mit dem Qualifier P (Pulse).

Beim Aufruf eines IEC-Schritts werden zuerst die deaktivierten Aktionen in alphabetischer Reihenfolge abgearbeitet; anschließend alle aktiven Aktionen, wiederum in alphabetischer Reihenfolge.

Um IEC-Schritte einfügen zu können, aktivieren Sie über den Menübefehl [Extras] / [IEC-Schritte benutzen] die Verwendung von IEC-Schritten. Alle neuen Schritte werden von nun an als IEC-Schritte eingefügt, bis Sie den Menübefehl erneut ausführen.

Im **Object Organizer** hängen die Aktionen direkt unter dem jeweiligen Baustein. Neue Aktionen können mit dem Menübefehl [Projekt] / [Aktion hinzufügen] ([Projekt] / [Add Action]) erzeugt werden.

Beispiel für AS-Baustein mit Aktionen im Object Organizer

20248ADE

Qualifier

Qualifier (Bestimmungszeichen) werden verwendet, um zu steuern, wie IEC-Aktionen und IEC-Schritt assoziieren.

Qualifier	Bedeutung	Beschreibung
N	**N**on-stored	Die Aktion ist so lange aktiv wie der Schritt.
R	Overriding **R**eset	Die Aktion wird deaktiviert.
S	**S**et (stored)	Die Aktion wird aktiviert und bleibt bis zu einem Reset aktiv.
L [1)]	Time **L**imited	Die Aktion wird für eine bestimmte Zeit aktiviert, maximal solange der Schritt aktiv ist.
D [1)]	Time **D**elayed	Die Aktion wird nach einer bestimmten Zeit aktiviert, sofern der Schritt aktiv ist. Die Aktion bleibt aktiv, solange der Schritt aktiv ist.
P	**P**ulse	Die Aktion wird genau einmal ausgeführt, wenn der Schritt aktiv wird.
SD [1)]	**S**tored and time **D**elayed	Die Aktion wird nach einer bestimmten Zeit aktiviert und bleibt bis zu einem Reset aktiv.
DS [1)]	Time **D**elayed and **S**tored	Die Aktion wird nach einer bestimmten Zeit aktiviert, sofern der Schritt aktiv ist, und bleibt bis zum Reset aktiv.
SL [1)]	**S**tored and time **L**imited	Die Aktion ist für eine bestimmte Zeit aktiv.

Tabelle 4: AS-Qualifier und ihre Bedeutung

1) Die Bestimmungszeichen L, D, SD, DS und SL benötigen eine Zeitangabe im `TIME`-Konstantenformat, z. B. L `T#5s`.

Hinweis

Wenn eine Aktion deaktiviert wird, wird sie **noch einmal** ausgeführt. Jede Aktion wird also mindestens zweimal ausgeführt. Das gilt auch für eine Aktion mit dem Qualifier `P` (Pulse).

Beim Aufruf eines IEC-Schritts werden zuerst die deaktivierten Aktionen in alphabetischer Reihenfolge abgearbeitet; anschließend alle aktiven Aktionen, wiederum in alphabetischer Reihenfolge.

Achtung

Wenn dieselbe Aktion in zwei unmittelbar aufeinanderfolgenden Schritten Zeit-Qualifier benutzt, kann der Zeit-Qualifier bei der zweiten Verwendung nicht mehr wirksam werden.

Um dies zu umgehen, muss ein Zwischenschritt eingefügt werden. In dem zusätzlich zu durchlaufenden Zyklus muss der Aktionszustand erneut initialisiert werden.

Implizite Variablen

In AS gibt es implizit deklarierte Variablen.

Zu jedem Schritt gehört ein Flag, welches den Zustand des Schritts speichert. Das Schritt-Flag (aktiver oder inaktiver Zustand des Schritts) heißt bei IEC-Schritten `<StepName>.x`, bei einfachen Schritten `<StepName>`. Diese boolesche Variable hat den Wert TRUE, wenn der zugehörige Schritt aktiv ist, und FALSE, wenn er inaktiv ist. Sie kann in jeder Aktion und Transition des AS-Bausteins benutzt werden.

Ob eine IEC-Aktion aktiv ist oder nicht, kann mit der Variablen `<AktionsName>.x` abgefragt werden.

Bei den IEC-Schritten kann mit der impliziten Variablen `<StepName>.t` die aktive Zeitdauer der Schritte abgefragt werden.

Auf implizite Variablen kann von anderen Programmen aus zugegriffen werden.

Beispiel:

```
BOOL_VAR1 := sfc.step1.x;
```

Dabei ist `step1.x` die implizite boolesche Variable, die den Zustand des IEC-Schritts `step1` im Baustein `sfc1` darstellt.

Flags

Zur Steuerung des Ablaufs können in AS Flags genutzt werden, die während der Projektabarbeitung automatisch erzeugt werden. Dazu müssen entsprechende Variablen global oder lokal und als Aus- oder Eingabevariable deklariert werden.

Beispiel: Wenn ein Schritt länger aktiv ist als in seinen Attributen angegeben, wird ein Flag gesetzt, das über eine Variable namens `SFCError` zugänglich wird (`SFCError` wird TRUE). Folgende Flag-Variablen sind möglich:

Flag-Variable	Typ	Beschreibung
SFCEnableLimit	BOOL	Wenn die Variable TRUE ist, werden Zeitüberschreitungen bei den Schritten in `SFCError` registriert. Ansonsten werden Zeitüberschreitungen ignoriert. Das Flag kann beispielsweise bei Inbetriebnahme oder Handbetrieb nützlich sein.
SFCInit	BOOL	Wenn die Variable TRUE ist, wird die Ablaufsprache auf den Init-Schritt zurückgesetzt. Die anderen AS-Flags werden ebenfalls zurückgesetzt (Initialisierung). Solange die Variable TRUE ist, bleibt der Init-Schritt gesetzt (aktiv), wird aber nicht ausgeführt. Erst wenn `SFCInit` wieder auf FALSE gesetzt wird, wird der Baustein normal weiterbearbeitet.
SFCReset	BOOL	Die Variable verhält sich ähnlich wie `SFCInit`. Im Unterschied zu dieser wird allerdings nach der Initialisierung der Init-Schritt weiter abgearbeitet. So könnte beispielsweise im Init-Schritt das Flag `SFCReset` gleich wieder auf FALSE gesetzt werden.
SFCQuitError	BOOL	Solange die Variable TRUE ist, wird die Abarbeitung des AS-Diagramms angehalten. Eine in der Variablen `SFCError` registrierte Zeitüberschreitung wird dabei zurückgesetzt. Wenn die Variable wieder auf FALSE gesetzt wird, werden alle bisherigen Zeiten in den aktiven Schritten zurückgesetzt. Voraussetzung dafür ist die Deklaration der Variablen `SFCError`, die die Zeitüberschreitung registriert.
SFCPause	BOOL	Solange die Variable TRUE ist, wird die Abarbeitung des AS-Diagramms angehalten.
SFCError	BOOL	Diese Variable wird TRUE, wenn in einem AS-Diagramm eine Zeitüberschreitung aufgetreten ist. Wenn nach der ersten Zeitüberschreitung eine weitere auftritt, wird diese nicht mehr registriert, es sei denn, die Variable `SFCError` wurde vorher wieder zurückgesetzt. Die Deklaration von `SFCError` ist Voraussetzung für das Funktionieren der anderen Flag-Variablen, die zur Kontrolle des zeitlichen Ablaufs dienen (`SFCErrorStep`, `SFCErrorPOU`, `SFCQuitError`, `SFCErrorAnalyzationTable`).
SFCTrans	BOOL	Die Variable wird TRUE, wenn eine Transition schaltet.
SFCErrorStep	STRING	Wenn in `SFCError` eine Zeitüberschreitung im AS-Diagramm registriert wird, speichert die Variable den Namen des Schritts, der die Zeitüberschreitung verursacht hat. Voraussetzung dafür ist die Deklaration der Variablen `SFCError`, die die Zeitüberschreitung registriert.
SFCErrorPOU	STRING	Wenn in `SFCError` eine Zeitüberschreitung im AS-Diagramm registriert wird, speichert die Variable den Namen des Bausteins, in dem die Zeitüberschreitung aufgetreten ist. Voraussetzung dafür ist die Deklaration der Variablen `SFCError`, die die Zeitüberschreitung registriert.
SFCCurrentStep	STRING	In dieser Variablen wird der Name des momentan akitven Schritts gespeichert, unabhängig von der Zeitüberwachung. Bei einer Paralellverzweigung wird der Name des Schritts im äußersten rechten Zweig gespeichert.

Tabelle 5: AS-Flags und ihre Bedeutung

Flag-Variable	Typ	Beschreibung
SFCErrorAnalyzati-onTable	ARRAY [0..15] OF ExpressionResult	Diese Flag-Variable gibt für jede Variable eines zusammengesetzten Transitionsausdrucks, die zu einem FALSE der Transition und damit zu einer Zeitüberschreitung im vorangehenden Schritt führt, folgende Informationen aus: Name, Adresse, Kommentar, aktueller Wert. Es können Informationen zu maximal 16 Variablen ausgegeben werden. Die Struktur `ExpressionResult` sowie die implizit verwendeten Analyse-Bausteine sind in der Bibliothek `AnalyzationNew.lib` enthalten. Die Analyse-Bausteine können auch explizit in Bausteinen verwendet werden, die nicht in SFC programmiert sind. Voraussetzung für die Analyse des Transitionsausdrucks ist die Registrierung einer Zeitüberschreitung im vorangehenden Schritt. Daher muss dort eine Zeitüberwachung implementiert sein und die FlagVariable `SFCError` im Baustein deklariert sein.
SFCTip, SFCTipMode	BOOL	Diese Variable erlaubt den Tip-Betrieb des SFC, der durch `SFCTipMode = TRUE` eingeschaltet wird. Es kann dann nur zum nächsten Schritt weitergeschaltet werden, indem `SFCTip` auf TRUE gesetzt wird. Solange `SFCTipMode` auf FALSE gesetzt ist, kann zusätzlich auch über die Transitionen weitergeschaltet werden.

Tabelle 5: AS-Flags und ihre Bedeutung

Alternativ-Zweig

Zwei oder mehr Zweige können als Alternativ-Verzweigungen definiert werden. Jeder Alternativ-Zweig muss mit einer Transition beginnen und enden. Alternativ-Verzweigungen können Parallel-Verzweigungen und weitere Alternativ-Verzweigungen enthalten. Eine Alternativ-Verzweigung beginnt an einer horizontalen Linie (Alternativ-Anfang) und endet an einer horizontalen Linie (Alternativ-Ende) oder mit einem Sprung.

Wenn der Schritt, der der Linie am Alternativ-Anfang vorangeht, aktiv ist, werden die Alternativ-Zweige von links nach rechts abgearbeitet und die jeweils erste Transition eines Alternativ-Zweigs ausgewertet. Die erste Transition, deren Transitionsbedingung den Wert TRUE hat, wird geöffnet und die nachfolgenden Schritte werden aktiviert.

Parallel-Zweig

Zwei oder mehr Verzweigungen in AS können als Parallel-Zweige definiert werden. Jeder Parallel-Zweig muss mit einem Schritt beginnen und enden. Parallel-Zweige können Alternativ-Verzweigungen oder weitere Parallel-Verzweigungen enthalten. Ein Parallel-Zweig beginnt an einer doppelten Linie (Parallel-Anfang) und endet an einer doppelten Linie (Parallel-Ende) oder bei einem Sprung. Eine Parallel-Verzweigung kann mit einer Sprungmarke versehen werden.

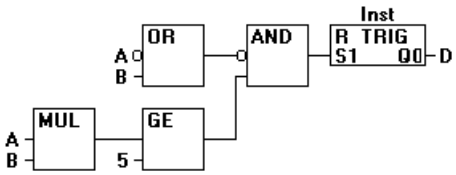
Wenn der Schritt, der der Linie am Parallel-Anfang vorangeht, aktiv ist, dann werden die Alternativverzweigungen von links nach rechts abgearbeitet und jeweils die erste Transistion ausgewertet. Die erste Transition, deren Transitionsbedingung den Wert TRUE hat, wird geöffnet und die nachfolgenden Schritte aktiviert.

2.4.5 Funktionsplan FUP (FBD)

FUP ist eine grafisch orientierte Programmiersprache. Sie arbeitet mit einer Liste von Netzwerken. Jedes Netzwerk enthält eine Struktur, die Folgendes darstellen kann:

- einen logischen bzw. arithmetischen Ausdruck
- den Aufruf eines Funktionsblocks
- einen Sprung
- eine RETURN-Anweisung

Beispiel für ein Netzwerk in FUP



20249ADE

2.4.6 Freigrafischer Funktionsplaneditor CFC (CFC)

CFC (Continuous Function Chart) basiert auf dem Funktionsplan, arbeitet jedoch nicht wie dieser mit Netzwerken, sondern mit frei platzierbaren Elementen. Dies erlaubt beispielsweise Rückkoppelungen.

Beispiel für eine Implementierung in CFC

20250ADE

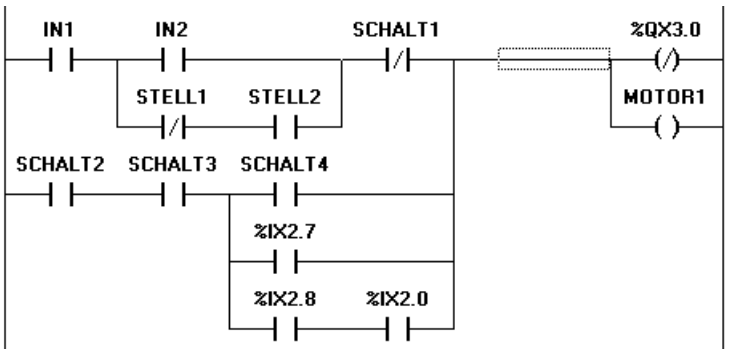
2.4.7 Kontaktplan KOP (LD)

KOP ist eine grafisch orientierte Programmiersprache, die dem Prinzip einer elektrischen Schaltung angenähert ist.

Mit KOP können Sie logische Schaltwerke konstruieren und Netzwerke erstellen. Daher können Sie KOP benutzen, um den Aufruf von anderen Bausteinen zu steuern.

KOP besteht aus einer Folge von Netzwerken. Ein Netzwerk wird auf der linken und rechten Seite von einer vertikalen Stromleitung begrenzt. Dazwischen befindet sich ein Schaltplan aus Kontakten, Spulen und Verbindungslinien.

Beispiel eines Netzwerks aus Kontakten und Spulen in KOP



20251ADE

Kontakt

Jedes Netzwerk besteht auf der linken Seite aus einer Folge von Kontakten (Symbol: **| |**), die von links nach rechts den Zustand "AN" oder "AUS" weitergeben. Diese Zustände entsprechen den booleschen Werten TRUE und FALSE.

Zu jedem Kontakt gehört eine boolesche Variable. Wenn diese Variable TRUE ist, wird der Zustand über die Verbindungslinie nach rechts weitergegeben, ansonsten erhält die rechte Verbindung den Wert "AUS".

Kontakte können parallel oder in Reihe geschaltet sein.

Bei der Parallelschaltung muss **einer** der Parallelzweige den Wert "An" übergeben, damit die Parallelverzweigung den Wert "An" weitergibt.

Bei der Reihenschaltung müssen alle Kontakte den Zustand "An" an den nächsten Kontakt übergeben, damit der letzte Kontakt den Zustand "An" weitergibt.

Dies entspricht einer elektrischen Parallel- bzw. Reihenschaltung.

Kontakte können negiert werden (Symbol: **|/|**). Bei einem negierten Kontakt wird der Wert der Linie weitergegeben, wenn die Variable FALSE ist.

Spule

Auf der rechten Seite eines Netzwerks in KOP befindet sich eine beliebige Anzahl so genannter Spulen (Symbol: **( )**).

Eine Spule gibt den Wert der Verbindungen nach rechts weiter und kopiert ihn in eine zugehörige boolesche Variable. An der Eingangslinie können die Werte "AN" (entspricht TRUE bei einer booleschen Variablen) oder "AUS" (entspricht FALSE bei einer booleschen Variablen) anliegen.

Spulen können nur parallel geschaltet werden.

Spulen können negiert werden (Symbol: **(/)**). Eine negierte Spule kopiert den negierten Wert einer Verbindung in die zugehörige boolesche Variable.

Set- und Reset-Spulen

Spulen können auch als Set- oder Reset-Spulen definiert sein.

Eine Set-Spule (Symbol: **(S)**) kann den Wert TRUE in der zugehörigen booleschen Variablen nicht überschreiben. Wenn die Variable einmal auf TRUE gesetzt wurde, dann bleibt sie es auch.

Eine Reset-Spule (Symbol: **(R)**) kann den Wert TRUE in der zugehörigen booleschen Variablen nicht überschreiben. Wenn die Variable einmal auf FALSE gesetzt wurde, dann bleibt sie es auch.

Funktionsblock, Programm

In KOP gibt es auch Programme und Funktionsblöcke. Diese müssen im Netzwerk einen Eingang und einen Ausgang mit booleschen Werten haben. Funktionsblöcke und Programme werden auf der linken Seite des KOP-Netzwerks verwendet.

KOP als FUP

Wenn Sie das Ergebnis einer Kontaktschaltung zur Steuerung anderer Bausteine einsetzen wollen, haben Sie dafür zwei Möglichkeiten:

- Sie legen das Ergebnis mit Hilfe der Spulen in einer globalen Variable ab, die an anderer Stelle benutzt werden kann.
- Sie bauen den eventuellen Aufruf direkt in Ihr KOP-Netzwerk ein. Dazu führen Sie einen Baustein mit EN-Eingang ein.

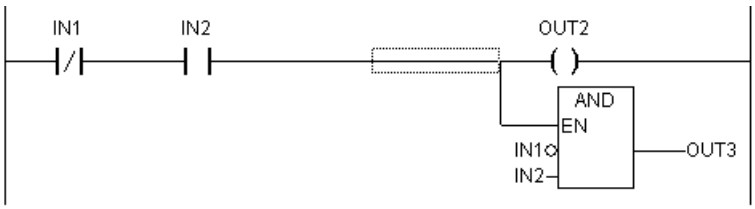
Bausteine mit EN-Eingang sind normale Operanden, Funktionen, Programme oder Funktionsblöcke, die einen zusätzlichen EN-Eingang haben. Der EN-Eingang ist immer vom Typ B00L. Der Baustein mit EN-Eingang wird ausgewertet, wenn EN den Wert TRUE hat.

Wenn am EN-Eingang der Wert FALSE angelegt wird, bleiben die Werte der Ausgänge von der letzten Auswertung mit dem Wert TRUE an EN erhalten.

Ein EN-Baustein wird parallel zu den Spulen geschaltet, wobei der EN-Eingang mit der Verbindungslinie zwischen den Kontakten und den Spulen verbunden wird. Wenn über diese Linie die Information "AN" transportiert wird, wird dieser Baustein ganz normal ausgewertet.

Ausgehend von einem solchen EN-Baustein können Netzwerke wie in FUP erstellt werden.

Beispiel eines KOP-Netzwerks mit einem EN-Baustein



20252ADE

2.5 Debugging

Dieses Kapitel gibt Ihnen eine kurze Einführung in die **Debugging**-Funktionen. Eine ausführliche Beschreibung der Debugging-Funktionen des PLC-Editor finden Sie auf Seite 178 (→ Kap. 4.6, Allgemeine Online-Funktionen).

Voraussetzungen

Die Debugging-Funktionen des PLC-Editor erleichtern Ihnen das Auffinden von Fehlern.

Um die Debugging-Funktionen nutzen zu können, führen Sie den Menübefehl [Projekt] / [Optionen] ([Project] / [Options]) aus. Im erscheinenden Dialog [Optionen] ([Options]) wechseln Sie in die Kategorie [Übersetzungsoptionen] ([Build]) und aktivieren die Option [Debugging] ([Debugging]) aus.

Breakpoint

Ein **Breakpoint** ist eine Stelle im Programm, an der die Abarbeitung angehalten wird. Somit ist es möglich, die Werte von Variablen an einer bestimmten Programmstelle zu betrachten.

Breakpoints können in allen Editoren gesetzt werden:

- in AWL und ST auf Zeilennummern
- in FUP und KOP auf Netzwerknummern
- in CFC auf Bausteine
- in AS auf Schritte

Hinweis:

In Funktionsblockinstanzen können keine Breakpoints gesetzt werden.

Einzelschritt

Durch schrittweise Abarbeitung können Sie die logische Korrektheit Ihres Programms überprüfen.

Einzelschritt bedeutet in den einzelnen Programmiersprachen:

- AWL: Das Programm bis zum nächsten `CAL-`, `LD` oder `JMP`-Befehl ausführen.
- ST: Die nächste Anweisung ausführen.
- FUP, KOP: Das nächste Netzwerk ausführen.
- AS: Die Aktion zum nächsten Schritt ausführen.
- CFC: Den nächsten Baustein (Box) im CFC-Programm ausführen.

Einzelzyklus

Beim Einzelzyklus wird die Abarbeitung nach jedem Zyklus angehalten.

Werte online ändern

Variablen können im laufenden Betrieb einmalig auf einen bestimmten Wert gesetzt werden (Wert schreiben) oder auch nach jedem Zyklus wieder neu mit einem bestimmten Wert beschrieben werden (**Forcen**).

Den Variablenwert können Sie im Online-Modus ändern, indem Sie auf den Variablenwert doppelklicken. Boolesche Variablen wechseln dadurch von TRUE auf FALSE bzw. umgekehrt. Bei alle anderen Variablentypen wird der Dialog [Variable xy schreiben] ([Write Variable xy]) geöffnet, in welchem Sie den aktuellen Variablenwert editieren können.

Monitoring

Im Online-Modus werden für alle Variablen, die auf dem Bildschirm sichtbar sind, laufend die aktuellen Werte aus der Steuerung gelesen und dargestellt. Diese Darstellung

finden Sie im Deklarations- und Programmeditor. Außerdem können Sie im Watch- und Rezepturverwalter und in einer Visualisierung aktuelle Variablenwerte ausgeben.

Zum Darstellen von Variablen aus Funktionsblockinstanzen muss die entsprechende Instanz geöffnet werden.

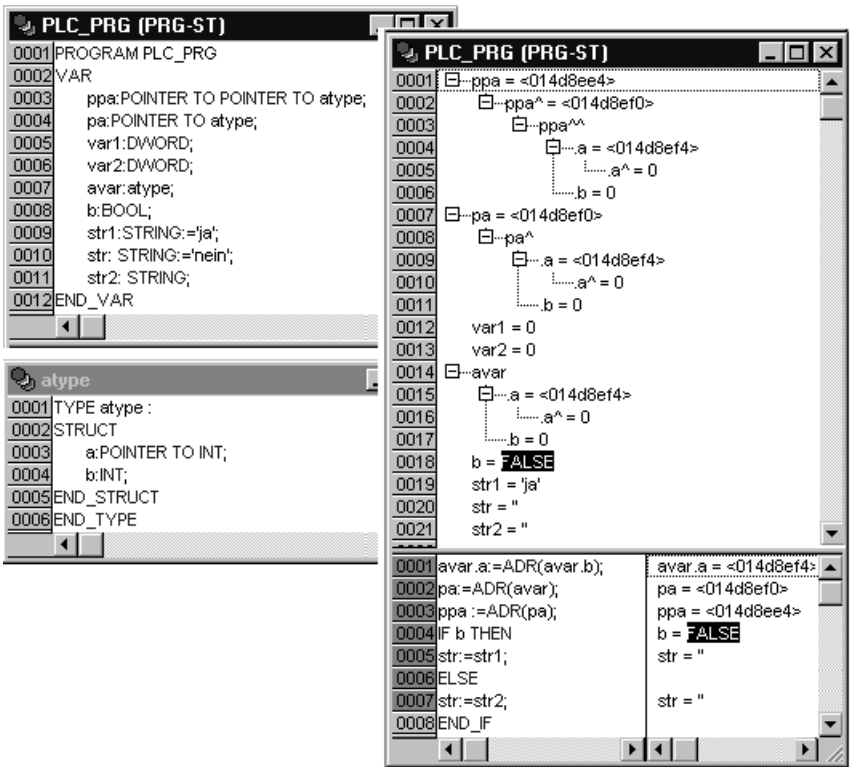
Bei VAR_IN_OUT-Variablen wird der dereferenzierte Wert ausgegeben.

Bei Pointern (Zeigern) wird im Deklarationsteil sowohl der Pointer als auch der dereferenzierte Wert ausgegeben. Im Programmteil wird nur der Pointer ausgegeben:

```
+ --pointervar = '<'pointervalue'>'
```

Pointer im dereferenzierten Wert werden ebenfalls entsprechend angezeigt. Mit einfachem Klick auf das Kreuz oder mit Doppelklick auf die Zeile wird die Anzeige expandiert bzw. kollabiert.

Beispiel für Monitoring von Pointern



20253ADE

In Implementierungen wird der Wert des Pointers angezeigt, für Dereferenzierungen der dereferenzierte Wert.

Monitoring von ARRAY-Komponenten

Über Konstanten und über Variablen indizierte ARRAY-Komponenten werden angezeigt:

```
anarray[1] = 5
anarray[i] = 1
```

Besteht der Index aus einem Ausdruck, wird die Komponente nicht angezeigt:

```
anarray[i+j]
anarray[i+1]
```

Hinweis

Die Anzahl der beim Monitoring darstellbaren Variablen ist begrenzt. Wenn diese Anzahl überschritten wird, steht anstelle des aktuellen Variablenwerts der Text "Zu viele Monitoring-Variablen".

Simulation

Bei der Simulation wird das erzeugte Steuerungsprogramm nicht in der Steuerung, sondern auf dem Rechner abgearbeitet, auf dem der PLC-Editor läuft. Es stehen alle Online-Funktionen zur Verfügung. Sie haben somit die Möglichkeit, die logische Korrektheit Ihres Programms ohne Steuerungshardware zu testen.

Hinweis:

Bausteine aus externen Bibliotheken laufen nicht in der Simulation.

Logbuch

Das Logbuch dient der Überwachung und der Fehlerrückverfolgung.

Das Logbuch zeichnet während des Online-Modus folgende Vorgänge chronologisch auf:

- Benutzeraktionen
- interne Vorgänge
- Statusänderungen
- Ausnahmezustände

2.6 Die Norm IEC 61131-3

Die Norm IEC 61131-3 ist ein internationaler Standard für Programmiersprachen von speicherprogrammierbaren Steuerungen.

Die im PLC-Editor realisierten Programmiersprachen sind konform den Anforderungen der Norm.

Nach diesem Standard besteht ein Programm aus folgenden Elementen:

- Strukturen
- Bausteine
- Globale Variablen

Die allgemeinen Sprachelemente werden in folgenden Abschnitten der Norm beschrieben:

- "Bezeichner"
- "Adressen"
- "Typen"
- "Kommentare"
- "Konstanten"

3 Programmierbeispiele

Dieses Kapitel beschreibt die Programmierung verschiedener Antriebsaufgaben unter Verwendung der im Handbuch "Bibliothek MPLCMotion_MDX für MOVI-PLC" beschriebenen Funktionsbausteine an konkreten Beispielen.

Außerdem führt Sie dieses Kapitel in den grundlegenden Umgang mit der Steuerungskonfiguration, dem Bibliotheksverwalter sowie dem [FUP-Editor] der Software MOVITOOLS®-MotionStudio ein.

Nähere Informationen entnehmen Sie bitte der Online-Hilfe der Software MOVITOOLS®-MotionStudio.

Detaillierte Beschreibungen zur Bibliothek `MPLCMotion_MDX` entnehmen Sie bitte dem Handbuch "Bibliothek MPLCMotion_MDX für MOVI-PLC".

3.1 Voraussetzungen

Um die erstellten Programme mit einer Motorachse testen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Die Steuerung MOVI-PLC® und ein Antriebsumrichter MOVIDRIVE® MDX60B/61B mit angeschlossenem Motor[1] sind gemäß den Anleitungen in den entsprechenden Handbüchern installiert.
- Zwischen dem CAN 1-Anschluss der Steuerung MOVI-PLC® und dem CAN 1-Anschluss des Antriebsumrichters MOVIDRIVE® MDX60B/61B ist eine Systembusverbindung gemäß den Hinweisen in den entsprechenden Handbüchern korrekt installiert[2].
- Der Engineering-PC ist gemäß den Hinweisen in den entsprechenden Handbüchern korrekt an die Steuerung MOVI-PLC® angeschlossen. Die Schnittstelle des Engineering-PCs ist entsprechend konfiguriert.
- Die Inbetriebnahme des Antriebsumrichters MOVIDRIVE® MDX60/B61B zur Ansteuerung durch eine MOVI-PLC®-Steuerung wurde mit Hilfe des Inbetriebnahme-Assistenten durchgeführt.

1) Zum Positionieren ist ein Motor mit Geber erforderlich.

2) Wenn die Steuerung MOVI-PLC® über den Rückwandstecker mit dem Antriebsumrichter kommuniziert, ist die Systembus-Verbindung über den CAN 1-Anschluss nicht erforderlich.

3.2 Positionieren einer Motorachse

Aufgaben-beschreibung

Wenn an einem digitalen Eingang des angeschlossenen Antriebsumrichters MOVIDRIVE® MDX60B/61B eine steigende Flanke der 24V DC Spannung auftritt, soll die angeschlossene Motorachse zehn Umdrehungen nach rechts gedreht und positioniert werden.

Die Bedienung des Antriebs erfolgt vollständig durch Verwendung der digitalen Eingänge. Die Steuerung MOVI-PLC® wertet die Eingänge des Antriebsumrichters aus und steuert die Positionierung der Motorachse.

Teilaufgaben

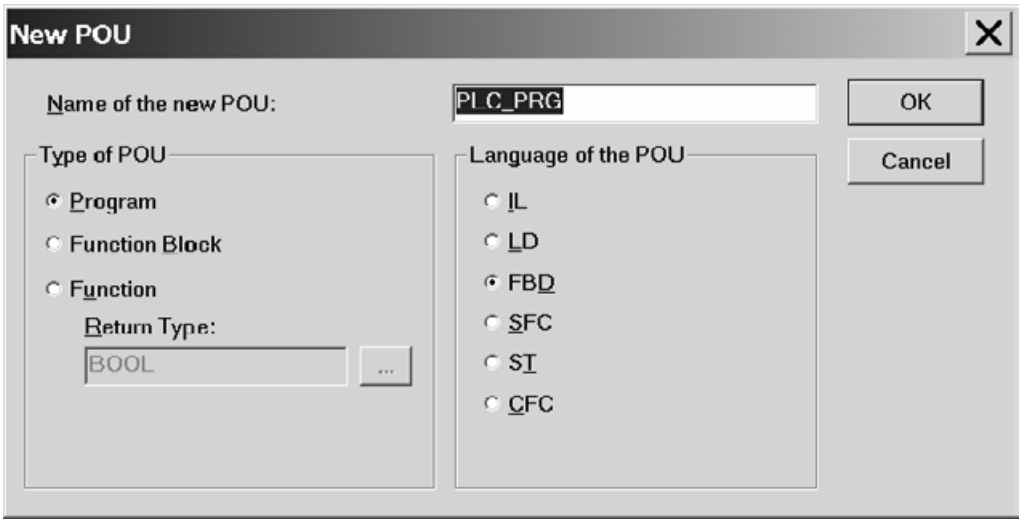
Das Programmierbeispiel untergliedert sich in folgende Teilaufgaben:

1. Erstellung eines neues Projekts
2. Einstellung der Steuerungskonfiguration
3. Einbinden der erforderlichen Bibliotheken
4. Programmierung der Kommunikation mit der Motorachse
5. Programmierung des Ein-/Ausschaltens des Antriebsumrichters
6. Programmierung des Positionierens der Motorachse
7. Übertragen des Projektes an die Steuerung MOVI-PLC®
8. Testen des Programms

Schritt 1

Neues Projekt erstellen

1. Schalten Sie den Engineering-PC und die Steuerung MOVI-PLC® [3] ein.
2. Starten Sie den PLC-Editor der Software MOVITOOLS®-MotionStudio gemäß der Anleitung im Kapitel "Start MOVITOOLS®-MotionStudio" im Handbuch der MOVI-PLC®.
3. Erstellen Sie ein neues Projekt, indem Sie auf [Datei] / [NEU] ([File] / [New]) klicken.
4. Wählen Sie im Dialog [Zielsystem Einstellungen] ([Target Settings]) die Konfiguration Ihrer MOVI-PLC®, in diesem Programmierbeispiel z.B. [MOVIPLC basic DHP11B], aus und klicken Sie auf die [OK]-Taste.



20030AXX

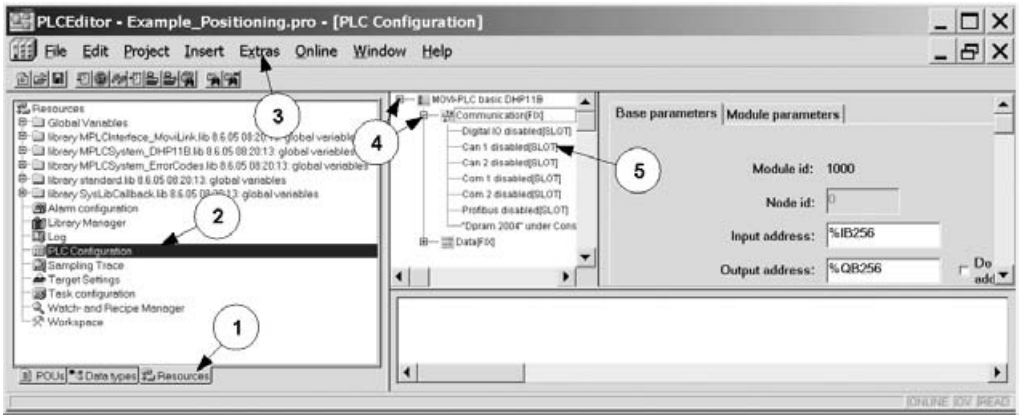
5. Verändern Sie für dieses Beispiel im Dialog [Neuer Baustein] ([New POU]) den Namen des Funktionsbausteins "PLC_PRG" nicht. Die Steuerung MOVI-PLC® führt den Baustein mit dem Namen "PLC_PRG" nach dem Programmstart (im Schritt 8 dieses Beispiels) automatisch aus.
6. Wählen Sie im Feld [Typ des Bausteins] ([Type of POU]) die Auswahl [Programm] ([Program]) aus.
7. Klicken Sie im Feld [Sprache des Bausteins] ([Language of the POU]) die Option [FUP] ([FBD]) an, und bestätigen Sie die Eingabe, indem Sie auf [OK] klicken.
8. Speichern Sie das Projekt, indem Sie auf [Datei] / [Speichern] ([File] / [Save]) klicken und den gewünschten Namen des Projekts eingeben. Wiederholen Sie das Speichern regelmäßig nach einigen ausgeführten Änderungen bzw. nach Abschluss der Programmerstellung.

3) Die Steuerung MOVI-PLC® muss entsprechend ihrer Ausführung mit Spannung versorgt werden, oder der Antriebsumrichter, in dem die Hardware der Steuerung MOVI-PLC® eingesteckt ist, muss eingeschaltet werden.

Schritt 2

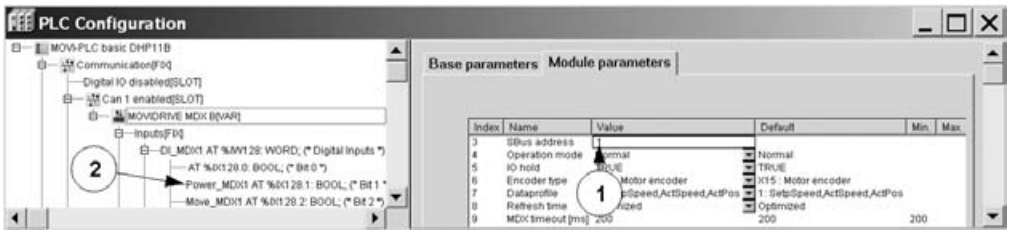
Steuerungskonfiguration einstellen

Um die Schnittstellen der Steuerung MOVI-PLC® für die Einbindung von Peripherie sowie die Kommunikation mit weiteren Geräten wie z.B. Umrichtern, E/A-Modulen oder auch einer übergeordneten Steuerung nutzen zu können, stellen Sie die Steuerungskonfiguration entsprechend ein.



20031AXX

1. Wählen Sie dazu das Fenster [Ressourcen] ([Recources]) aus, indem Sie auf den entsprechenden Tabreiter klicken [1].
2. Aktivieren Sie die [Steuerungskonfiguration] ([PLC Configuration]) durch einen Doppelklick [2].
3. Stellen Sie im Menü [Extras] die [Standardkonfiguration] ([Standard configuration]) einmalig ein und bestätigen Sie die erscheinende Abfrage, ob Sie die aktuelle Konfiguration verwerfen und mit der Standardkonfiguration ersetzen möchten, mit [Ja] [3].
4. Öffnen Sie den Konfigurationsbaum, indem Sie jeweils auf das [+]-Symbol vor [MOVI-PLC basic DHP11B] und [Communication] klicken [4].
5. Aktivieren Sie die CAN 1-Schnittstelle der Steuerung MOVI-PLC®, indem Sie mit der rechten Maustaste auf das Element [Can 1 disabled] klicken und im Kontextmenü [Element ersetzen] ([Replace element]) auswählen [5].
6. Konfigurieren Sie den Antriebsumrichter MOVIDRIVE® MDX60B/61B an der CAN 1-Schnittstelle, indem Sie mit der rechten Maustaste auf das Element [Can-1 enabled] klicken und im Kontextmenü [Unterelement anhängen] ([Append Subelement]) und [MOVIDRIVE MDX B] auswählen.



20032AXX

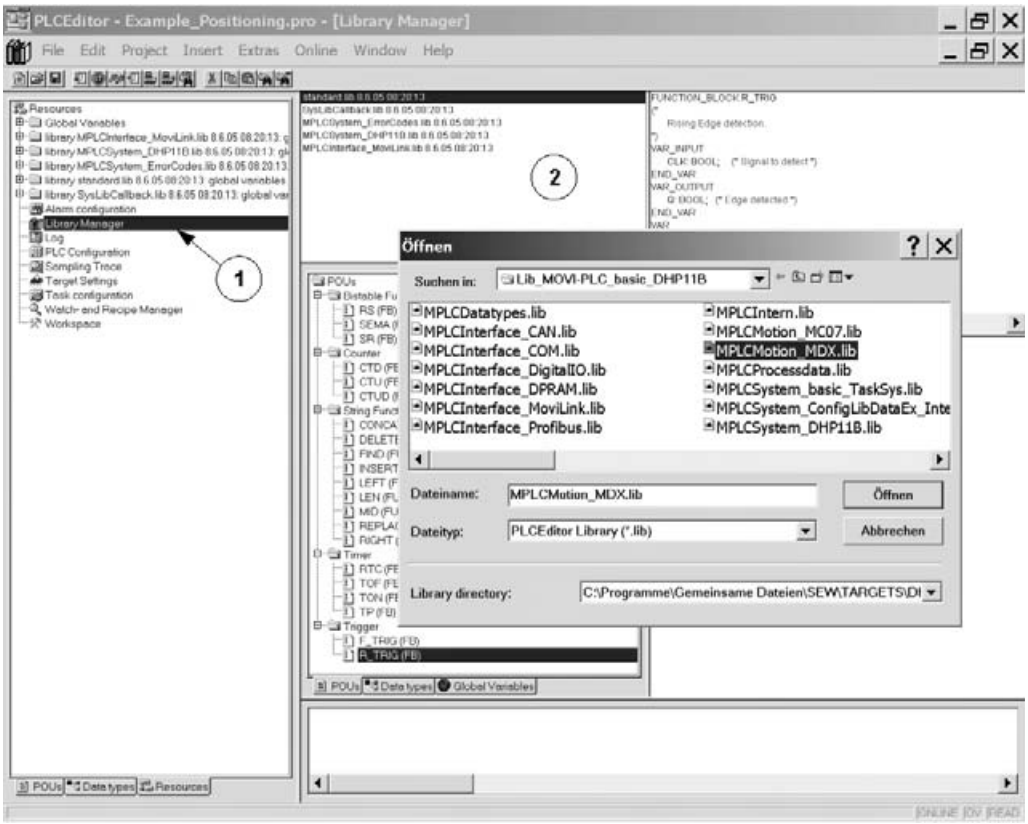
7. Markieren Sie das Element [MOVIDRIVE MDX B] und geben Sie im Fenster [Modulparameter] ([Module parameters]) die bei der Inbetriebnahme des Antriebsumrichters ausgewählte S-Bus-Adresse ein [1].
8. Im Steuerungsprogramm können Sie die Ein-/Ausgänge durch direkte Adressangabe mittels der Syntax "%I" / "%Q" verwenden.

   Einfacher ist jedoch die Nutzung symbolischer Bezeichner wie folgt: Klicken Sie jeweils auf das [+]-Symbol vor [MOVIDRIVE MDX B] und [Inputs]. Klicken Sie auf das entsprechende Feld [AT] im Konfigurationsbaum und geben den gewünschten Namen ein. In diesem Beispiel sind dies für die digitalen Eingänge des Antriebsumrichters der symbolische Bezeichner [DI_MDX1] bzw. für die einzelnen Ein-/Ausgänge die symbolischen Bezeichner [Power_MDX1] und [Move_MDX1] [2].

Schritt 3

Bibliotheken einbinden

Um die Funktionsbausteine der Bibliothek `MPLCMotion_MDX.lib` verwenden zu können, fügen Sie die Bibliothek [MPLCMotion_MDX.lib] gemäß der folgenden Beschreibung zu den bereits vorhandenen Bibliotheken hinzu.



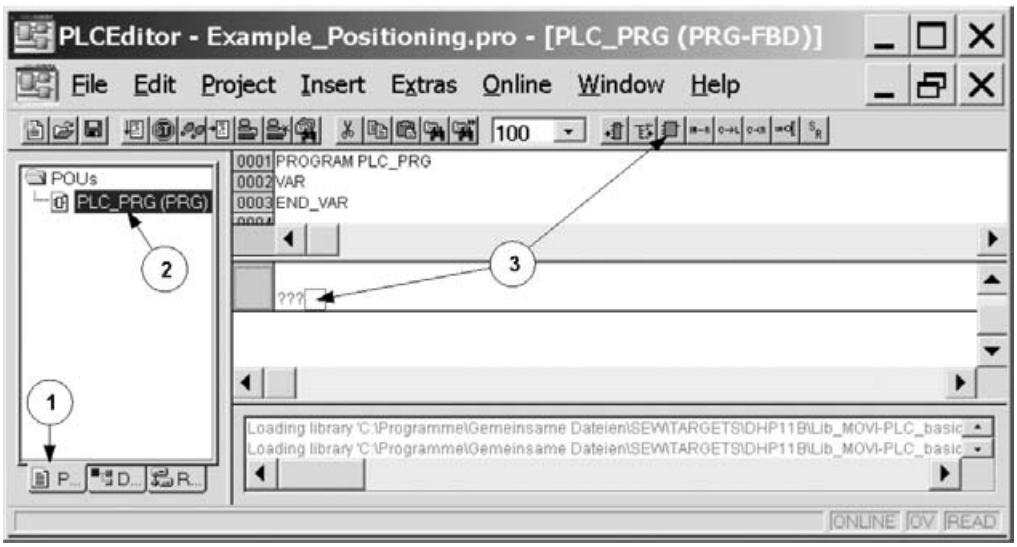
20033AXX

1. Aktivieren Sie den [Bibliotheksverwalter] ([Library Manager]) durch einen Doppelklick [1].
2. Klicken Sie mit der rechten Maustaste in das Bibliotheks-Feld und wählen Sie den Menüpunkt [Weitere Bibliothek] ([Additional Library]) aus [2].
3. Wählen Sie die Bibliothek [`MPLCMotion_MDX.lib`] aus und klicken Sie auf die Schaltfläche [Öffnen].

Schritt 4

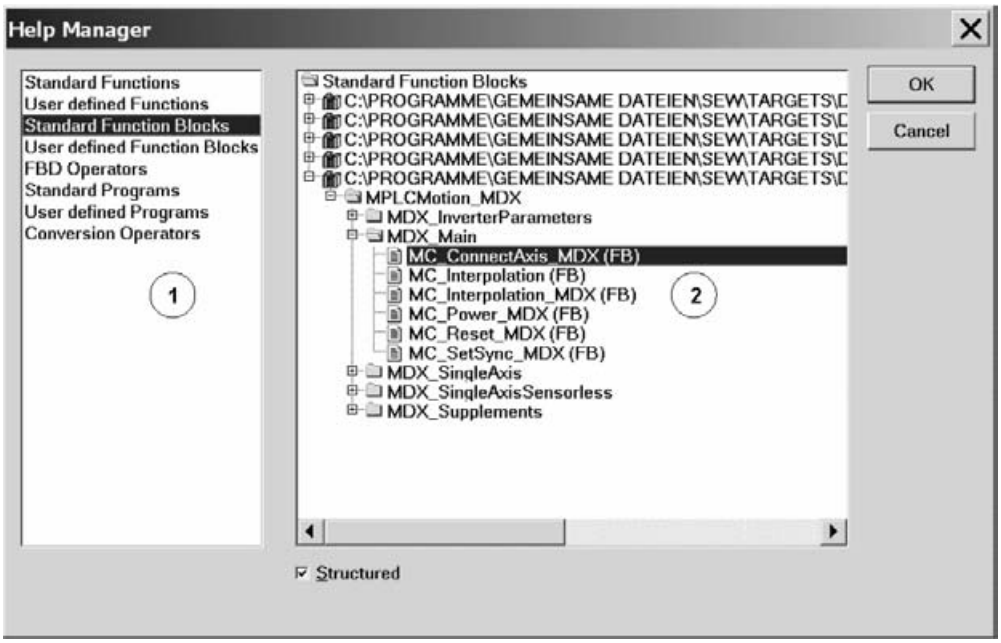
Programmierung der Kommunikation mit der Motorachse

Zum Aufbau und zur Durchführung der Kommunikation zwischen der Steuerung MOVI-PLC® und dem Antriebsumrichter fügen Sie eine Instanz des Funktionsbausteins MC_ConnectAxis_MDX wie folgt ein.



20034AXX

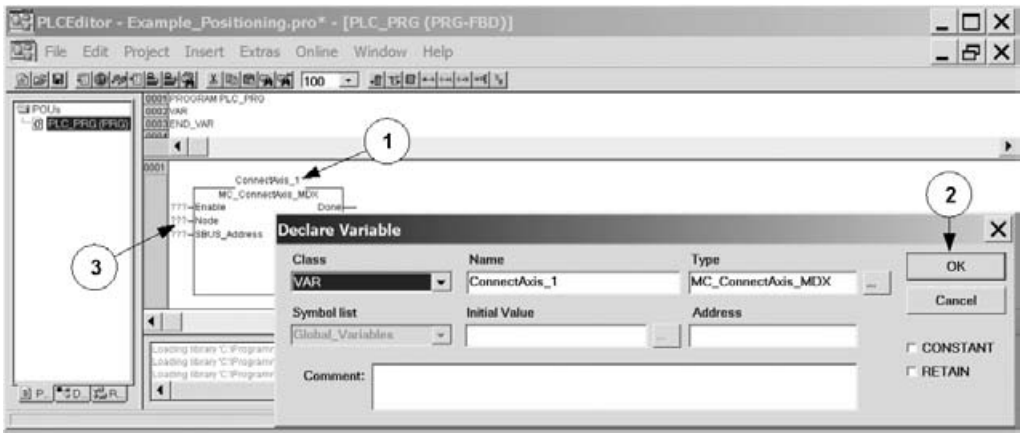
1. Wählen Sie dazu das Fenster [Bausteine] ([POUs]) aus, indem Sie auf den entsprechenden Tabreiter klicken [1].
2. Öffnen Sie den Editor des Bausteins "PLC_PRG(PRG)" durch einen Doppelklick [2].
3. Fügen Sie einen neuen Funktionsbaustein ein, indem Sie im ersten Netzwerk zunächst auf das Kästchen neben den Fragezeichen "???" und anschließend auf die [ ]-Schaltfläche klicken [3].



20036AXX

4. Markieren Sie den Text "AND" im neu eingefügten Funktionsbaustein.
5. Drücken Sie die <F2>-Taste. Die Software öffnet den Dialog [Eingabehilfe] ([Help Manager]).

6. Wählen Sie auf der linken Seite des Dialogs [Standard-Funktionsblöcke] ([Standard Function Blocks]) aus [1].
7. Wählen Sie auf der rechten Seite des Dialogs den Funktionsbaustein [MC_ConnectAxis_MDX (FB)] im Verzeichnis [MDX_Main] der Bibliothek [MPLCMotion_MDX] aus. Klicken Sie auf die [OK]-Schaltfläche [2].



20037AXX

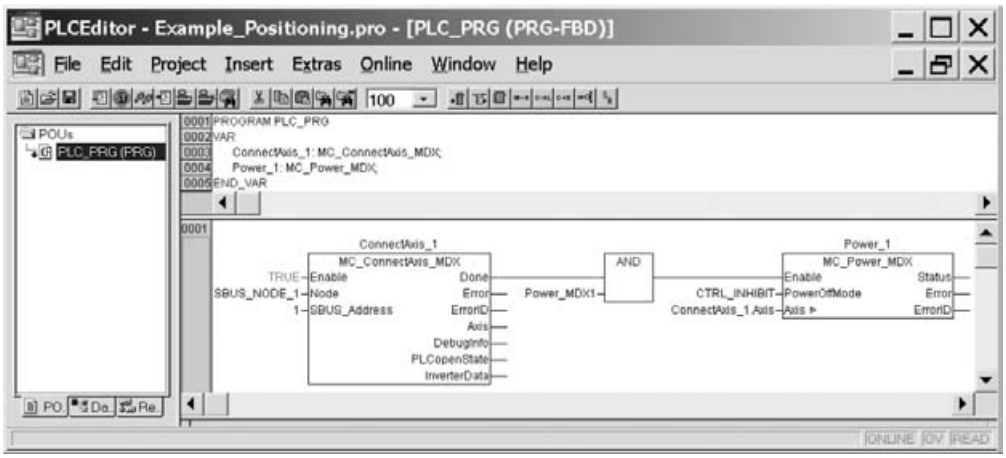
8. Klicken Sie auf den eingefügten Funktionsbaustein MC_ConnectAxis_MDX und anschließend auf die Fragezeichen "???" über dem Funktionsbaustein [1].
9. Geben Sie den Instanznamen (z.B. "ConnectAxis_1") des Funktionsbausteins ein [2] und drücken Sie die <Eingabe>-Taste. Bestätigen Sie den erscheinenden Dialog [Variablendeklaration] ([Declare Variable]), indem Sie auf die [OK]-Schaltfläche klicken.
10. Geben Sie an den Eingängen des Funktionsbausteins folgende Werte ein, indem Sie jeweils auf die Fragezeichen "???" links neben dem Eingang klicken, den Wert eingeben und die <Eingabe>-Taste drücken:

Enable	*TRUE*
Node	*SBUS_NODE_1*
SBUS_Address	bei der Inbetriebnahme des Antriebsumrichters eingestellte SBUS 1-Adresse

Schritt 5

Programmierung des Ein-/Ausschaltens des Antriebsumrichters

Fügen Sie einen Funktionsbaustein ein, der den Antriebsumrichter MOVIDRIVE® MDX60B/61B ein- bzw. ausschaltet. Das Einschalten ist nur möglich, wenn zuvor der Funktionsbaustein MC_ConnectAxis_MDX für diese Achse erfolgreich ausgeführt wurde. Außerdem soll der Antriebsumrichter nur eingeschaltet sein, wenn am digitalen Eingang *DI01* des Antriebsumrichters die 24V DC Spannung anliegt. Deshalb werden die entsprechenden Signale über einen "AND"-Baustein miteinander verknüpft.



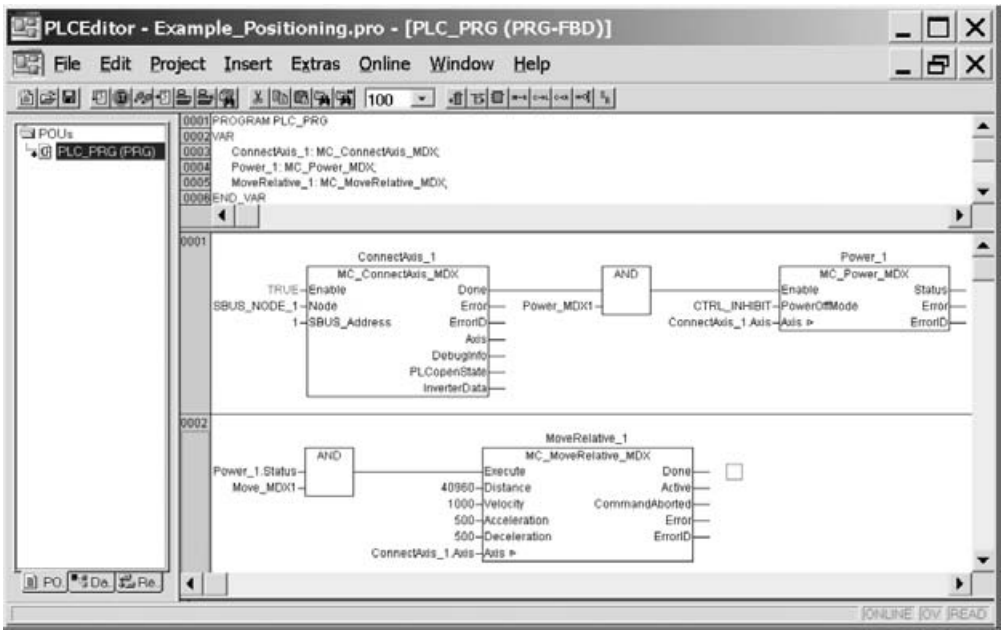
20038AXX

1. Klicken Sie dazu auf die Linie des Ausgangs *Done* des Funktionsbausteins MC_ConnectAxis_MDX.
2. Fügen Sie einen neuen "AND"-Funktionsbaustein ein, indem Sie auf die [ ] -Schaltfläche klicken.
3. Legen Sie am zweiten Eingang des neuen "AND"-Funktionsbausteins den am digitalen Eingang *DI01* des Antriebsumrichters eingelesenen Wert an, indem Sie den in der Steuerungskonfiguration gewählten symbolischen Namen (in diesem Beispiel "MDX1_DI.1" oder direkt "Power_MDX1") eingeben.
4. Fügen Sie den Funktionsbaustein MC_Power_MDX ein, indem Sie direkt rechts neben den "AND"-Funktionsbaustein und anschließend auf die [ ] -Schaltfläche klicken. Wandeln Sie den neu eingefügten "AND"-Funktionsbaustein gemäß der in Schritt 4 beschriebenen Vorgehensweise in einen Funktionsbaustein MC_Power_MDX um. Geben Sie dem Funktionsbaustein z.B. den Instanznamen "Power_1".
5. Geben Sie am Eingang *PowerOffMode* die Konstante "CTRL_INHIBIT" ein.
6. Geben Sie am Eingang *Axis* die Achsreferenz "ConnectAxis_1.Axis" ein, die von der Baustein-Instanz *ConnectAxis_1* ausgegeben wird. Klicken Sie dazu auf das Feld "???" vor dem Eingang *Axis* und geben Sie "ConnectAxis_1." ein. Wählen Sie den Eintrag [Axis] im Dialog aus, der automatisch nach Eingabe des Punktes erscheint. Bestätigen Sie die Eingabe, indem Sie die <Eingabe>-Taste drücken.

Schritt 6

Programmierung des Positionierens einer Motorachse

Fügen Sie einen Funktionsbaustein MC_MoveRelative_MDX ein, der eine relative Positionierbewegung der Motorachse steuert. In diesem Beispiel soll sich die Motorachse bei jeder steigenden Flanke der 24V DC Spannung am digitalen Eingang *DI02* des Antriebsumrichters zehn Umdrehungen (=10 x 4096 Inkremente der Gebers) im Uhrzeigersinn drehen. Während der Konstant-Fahrt soll sich die Motorachse mit einer Geschwindigkeit von 1000 1/min drehen.



20039AXX

1. Klicken Sie mit der rechten Maustaste in eine freie Fläche des Netzwerks [0001] und wählen Sie im erscheinenden Kontextmenü [Netzwerk (danach)] ([Network (after)]) aus.
2. Fügen Sie gemäß der im Schritt 4 beschriebenen Vorgehensweise im neuen Netzwerk [0002] einen Funktionsbaustein MC_MoveRelative_MDX ein und geben Sie ihm einen Instanznamen (z.B. "MoveRelative_1"). Den Funktionsbaustein MC_MoveRelative_MDX finden Sie im Verzeichnis [`MDX_SingleAxis`] der Bibliothek [`MPLCMotion_MDX`] unter [Standard Function Blocks].
3. Geben Sie an den Eingängen des Funktionsbausteins folgende Werte ein:

Distanz	*40960*
Velocity	*1000*
Acceleration	*500*
Deceleration	*500*
Axis	*ConnectAxis_1.Axis*

4. Der Fahrbefehl ist nur ausführbar, wenn der Funktionsbaustein MC_Power_MDX zuvor erfolgreich ausgeführt wurde. Deshalb legen Sie am Eingang *Execute* des Funktionsbausteins MC_MoveRelative_MDX das Ergebnis einer "AND"-Verknüpfung zwischen dem *Status*-Ausgang des Funktionsbausteins MC_Power_MDX und dem digitalen Eingang, der zum Starten der Bewegung vorgesehenen ist, an.

   Klicken Sie dazu auf die Linie vor dem Eingang *Execute*. Fügen Sie einen neuen "AND"-Funktionsbaustein ein, indem Sie auf die [ ]-Schaltfläche klicken. Belegen Sie die Eingänge des "AND"-Funktionsbausteins entsprechend (siehe Abbildung oben).
5. Zum Übersetzen des Projekts wählen Sie den Menüpunkt [Projekt] ([Project]) [Alles

übersetzen] ([Rebuild all]) aus. Bei fehlerfreier Programmierung zeigt das Meldefenster "0 Error(s), 0 Warnings(s)" an.

Schritt 7

Projekt an die Steuerung MOVI-PLC® übertragen

Klicken Sie auf den Menüpunkt [Online] / [Kommunikationsparameter] ([Communication Parameters]).

Stellen Sie im erscheinenden Dialog die Kommunikationsparameter entsprechend des verwendeten Kommunikationskanals ein. Dieser Schritt ist nur einmal erforderlich.

Klicken Sie anschließend auf den Menüpunkt [Online] / [Einloggen] ([Online] / [Login]).

Bestätigen Sie die im erscheinenden Dialog angezeigte Frage, ob ein Programm geladen werden soll, indem Sie auf [Ja] klicken.

Schritt 8

Programm testen

Im letzten Schritt führen Sie das Programm aus, indem Sie die verwendeten digitalen Eingänge entsprechend schalten.

Warnung:

In Abhängigkeit von der Klemmenbelegung, des Antriebsumrichter-Zustandes und des Steuerungsprogramms kann sich die Motorachse nach dem Starten der Steuerung MOVI-PLC® sofort bewegen. Um Schäden zu vermeiden, halten Sie ausreichend Abstand von allen bewegten Teilen!

Schalten Sie den Antriebsumrichter MOVIDRIVE® MDX60B/61B ein, wenn dies nicht bereits in Schritt 1 zum Einschalten der Steuerung MOVI-PLC® erfolgt ist.

Starten Sie die Steuerung MOVI-PLC®, indem Sie auf den Menüpunkt [Online] / [Start] ([Run]) klicken.

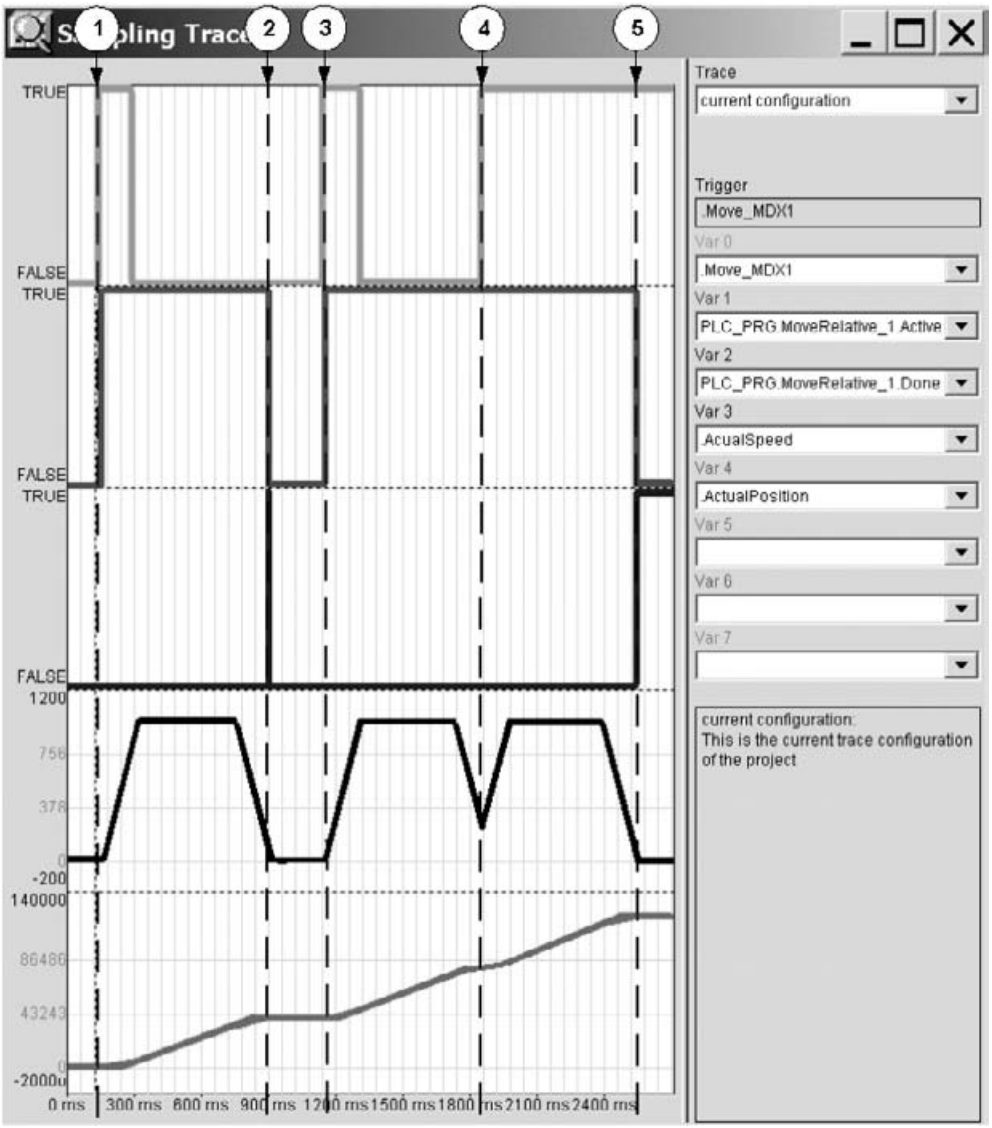
Schalten Sie den Antriebsumrichter ein, indem Sie nacheinander am Eingang *DI00* "/ Reglersperre" und am Eingang *DI01* (Enable des Funktionsbausteins MC_Power_MDX) die 24V DC Spannung anlegen.

Starten Sie die Positionierbewegung der Motorachse, indem Sie am Eingang *DI02* des Antriebsumrichters die 24V DC Spannung anlegen.

Die Programmierung war erfolgreich, wenn sich die Motorachse mit jeder steigenden Flanke der 24V DC Spannung am Eingang *DI02* um zehn Umdrehungen in positiver Richtung dreht.

Weitere Details zum Verhalten der Steuerung MOVI-PLC® und des angeschlossenen Antriebsumrichters MOVIDRIVE® in diesem Beispiel entnehmen Sie bitte dem folgenden Abschnitt "Traceaufzeichnung".

Traceaufzeichnung



20087AXX

Bei der steigenden Flanke des Signals *Move_MDX1* am Eingang *Execute* des Funktionsbausteins MC_MoveRelative_MDX setzt sich die Motorachse zum Erreichen der Zielposition in Bewegung [1]. Der Ausgang *Active* zeigt dies an, indem er auf *TRUE* gesetzt wird.

Nach erfolgreicher Positionierung setzt der Funktionsbaustein den Ausgang *Active* wieder auf *FALSE* zurück und setzt den Ausgang *Done* auf *TRUE* [2]. In diesem Beispiel ist der Ausgang *Done* nur während eines Steuerungszyklus auf *TRUE* gesetzt, weil das Eingangssignal *Execute* bereits vor Abschluss der Positionierung auf *FALSE* zurückgesetzt wurde.

Aufgrund der anschließenden steigenden Flanke am Eingang *Execute* startet die Motorachse erneut eine Positionierbewegung [3].

Allerdings erfolgt die nächste steigende Flanke am Eingang *Execute* bereits vor Abschluss der Positionierung. Eine weitere Positionierung, ausgehend von der aktuellen Position der Motorachse zum Zeitpunkt der letzten steigenden Flanke am Eingang *Execute*, wird fließend angehängt [4]. Somit wird die Motorachse nicht auf Stillstand abgebremst, sondern direkt in die nächste Bewegung übergeführt.

Nach Abschluss der Positionierung setzt der Funktionsbaustein den Ausgang *Active* wieder auf *FALSE* zurück. Das Ausgangssignal *Done* bleibt in diesem Fall jedoch auch nach Abschluss der Positionierung auf TRUE gesetzt, weil das Eingangssignal Execute noch nicht auf FALSE zurückgesetzt wurde [5].

3.3 Drehzahlregelung einer Motorachse mit Geber

Aufgaben-beschreibung

Wenn an einem digitalen Eingang des angesteuerten Antriebsumrichters MOVIDRIVE® MDX60B/61B eine steigende Flanke der 24V DC Spannung auftritt, soll die angeschlossene Motorachse eine drehzahlgeregelte Bewegung starten. Mit Hilfe eines weiteren digitalen Eingangs soll die Drehzahl der Motorachse zwischen zwei Werten umschaltbar sein. Zwei Eingänge sollen zum Starten einer Bremsbewegung mit Hilfe des Funktionsbausteins MC_Stop_MDX bzw. MC_AxisStop_MDX genutzt werden.

Die Bedienung des Antriebs erfolgt vollständig durch Verwendung der digitalen Eingänge. Die Steuerung MOVI-PLC® wertet die Eingänge des Antriebsumrichters aus und steuert die Drehzahlregelung der Motorachse.

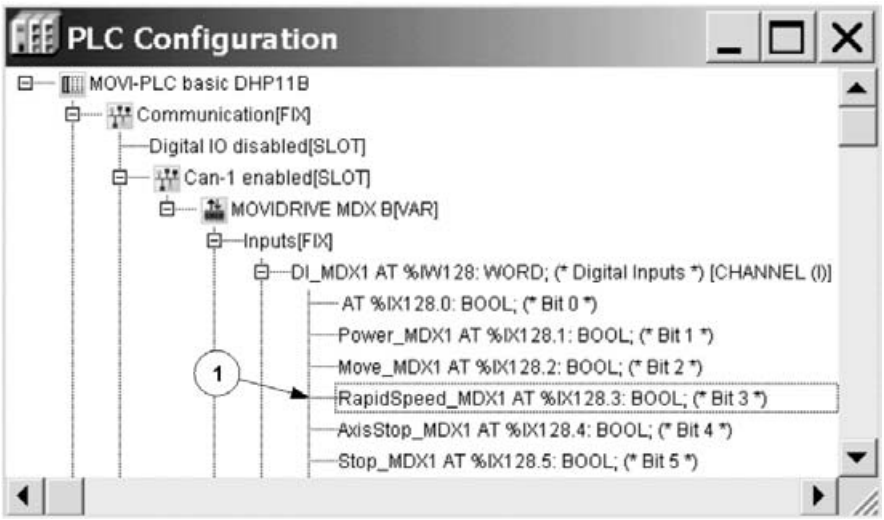
Programmierung

Unveränderte Schritte

Programmieren Sie die Schritte 1, 3-5 und 7 unverändert gemäß dem vorhergehenden Programmierbeispiel "Positionieren einer Motorachse".

Schritt 2

Steuerungskonfiguration erstellen


PLC Configuration
MOVI-PLC basic DHP11B
Communication[FIX]
Digital IO disabled[SLOT]
Can-1 enabled[SLOT]
MOVIDRIVE MDX B[VAR]
Inputs[FIX]
DI_MDX1 AT %IW128: WORD; (* Digital Inputs *) [CHANNEL (I)]
AT %IX128.0: BOOL; (* Bit 0 *)
Power_MDX1 AT %IX128.1: BOOL; (* Bit 1 *)
Move_MDX1 AT %IX128.2: BOOL; (* Bit 2 *)
RapidSpeed_MDX1 AT %IX128.3: BOOL; (* Bit 3 *)
AxisStop_MDX1 AT %IX128.4: BOOL; (* Bit 4 *)
Stop_MDX1 AT %IX128.5: BOOL; (* Bit 5 *)
1

20088AXX

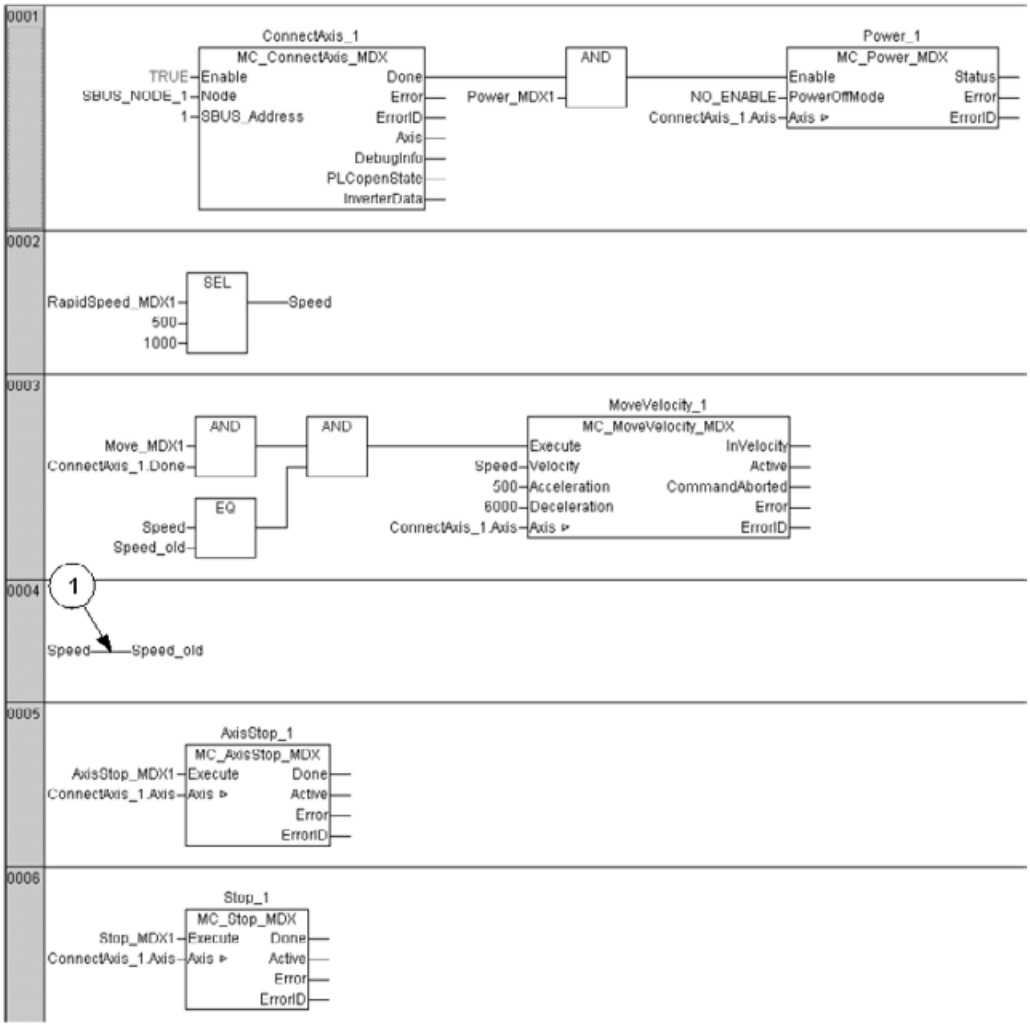
Ordnen Sie in der Steuerungskonfiguration zusätzlich zu den Bezeichnern [Power_MDX1] und [Move_MDX1] auch die Bezeichner

- [RapidSpeed_MDX1]
- [AxisStop_MDX1]
- [Stop_MDX1]

den digitalen Eingängen des Antriebsumrichters MOVIDRIVE® MDX60B/61B gemäß der Abbildung zu [1].

Schritt 6

Programmierung der Drehzahlregelung



20089AXX

1. Erstellen Sie gemäß der beschriebenen Vorgehensweise im vorhergehenden Programmierbeispiel das in der Abbildung gezeigte Programm.
2. Zur Programmierung der Wertzuweisung der Variablen *Speed* zur Variablen *Speed_old* klicken Sie nach dem Einfügen des Netzwerks [0004] zunächst auf das Kästchen neben den Fragezeichen "???" und anschließend auf die [IN-A]-Schaltfläche [1].

   Ersetzen Sie die Fragezeichen "???" durch die Variablennamen.

Schritt 8

Programm testen

Führen Sie das Programm aus, indem Sie die verwendeten digitalen Eingänge entsprechend schalten.

Warnung:

In Abhängigkeit von der Klemmenbelegung, des Antriebsumrichter-Zustandes und des Steuerungsprogramms kann sich die Motorachse nach dem Starten der Steuerung MOVI-PLC® sofort bewegen. Um Schäden zu vermeiden halten Sie ausreichend Abstand von allen bewegten Teilen!

Schalten Sie den Antriebsumrichter MOVIDRIVE® MDX60B/61B ein, wenn dies nicht bereits in Schritt 1 zum Einschalten der Steuerung MOVI-PLC® erfolgt ist.

Starten Sie die Steuerung MOVI-PLC®, indem Sie auf den Menüpunkt [Online] / [Start] ([Run]) klicken.

Schalten Sie den Antriebsumrichter ein, indem Sie nacheinander am Eingang *DI00* "/ Reglersperre" und am Eingang *DI01* (*Enable* des Funktionsbausteins MC_Power_MDX) die 24V DC Spannung anlegen.

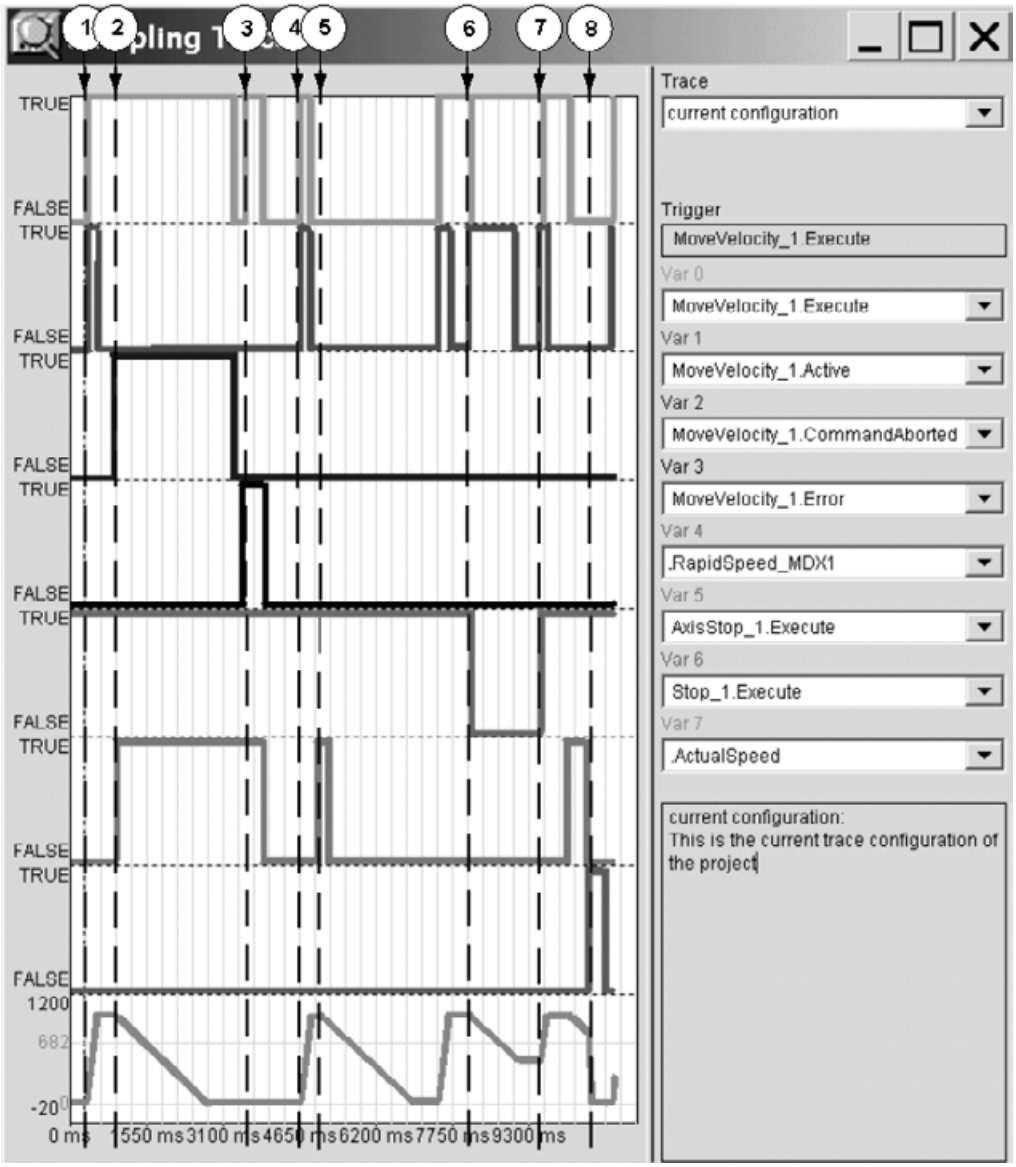
Starten Sie die Drehzahlregelung der Motorachse, indem Sie am Eingang *DI02* des Antriebsumrichters 24V DC Spannung anlegen.

Die Programmierung war erfolgreich, wenn

- sich die Motorachse mit einer steigenden Flanke der 24V DC Spannung am Eingang DI02 zu drehen beginnt,
- die Drehzahl der Motorachse durch den Wechsel zwischen 0V und 24V DC Spannung am Eingang *DI03* zwischen 500 1/min und 1000 1/min umschaltet und
- die Bewegung der Motorachse durch Anlegen der 24V DC Spannung an den Eingang *DI04* oder *DI05* abgebremst wird.

Weitere Details zum Verhalten der Steuerung MOVI-PLC® und des angeschlossenen Antriebsumrichters MOVIDRIVE® in diesem Beispiel entnehmen Sie bitte dem folgenden Abschnitt "Traceaufzeichnung".

Traceaufzeichnung



20090AXX

Bei steigender Flanke am Eingang *Execute* des Funktionsbausteins MC_MoveVelocity_MDX startet die Motorachse die Drehzahlregelung [1]. Der Funktionsbaustein setzt den Ausgang *Active* nur solange auf *TRUE*, bis die Solldrehzahl erreicht ist. Bei Erreichen der Solldrehzahl wird der Ausgang *Active* auf *FALSE* zurückgesetzt und der Ausgang *InVelocity* (der im Zeitdiagram nicht aufgezeichnet ist) wird auf *TRUE* gesetzt.

Aufgrund der steigenden Flanke am Eingang *Execute* des Funktionsbausteins MC_AxisStop_MDX führt die Motorachse eine Bremsbewegung mit der am Eingang *Deceleration* des Funktionsbausteins MC_MoveVelocity_MDX spezifizierten Bremsbeschleunigung aus [2]. Den Abbruch der gerade durchgeführten Drehzahlregelung zeigt der Funktionsbaustein MC_MoveVelocity_MDX an, indem er den Ausgang *CommandAborted* auf *TRUE* setzt.

Während der Eingang *Execute* des Funktionsbausteins MC_AxisStop_MDX noch auf *TRUE* gesetzt ist, erfolgt eine erneute steigende Flanke am Funktionsbaustein MC_MoveVelocity_MDX. Obwohl die Bremsbewegung bereits abgeschlossen war, setzt sich die Motorachse nicht in Bewegung. Für eine erneute Bewegung muss zu-

nächst der Zustand "Stopping" verlassen werden, indem der Eingang *Execute* des Funktionsbausteins MC_AxisStop_MDX auf *FALSE* zurückgesetzt wird. Sobald die Bedingung erfüllt ist, startet die Motorachse bei steigender Flanke am Eingang *Execute* des Funktionsbausteins MC_MoveVelocity_MDX wieder die Bewegung mit Drehzahlregelung [4].

Bei der anschließenden Flanke am Eingang *Execute* des Funktionsbausteins MC_AxisStop_MDX wird wieder eine Bremsbewegung gestartet [5]. Allerdings setzt in diesem Fall der Funktionsbaustein MC_MoveVelocity_MDX den Ausgang *CommandAborted* nicht auf *TRUE*, weil der Eingang *Execute* bereits zuvor auf *FALSE* zurückgesetzt wurde.

Beim Umschalten des Signals *RapidSpeed_MDX1* von *TRUE* auf *FALSE* wird der Eingang *Execute* des Funktionsbausteins MC_MoveVelocity_MDX auf *FALSE* zurückgesetzt [6]. Das Zurücksetzen wird durch den Vergleichs-Baustein [EQ] im Steuerungsprogramm während eines Steuerungszyklus bewirkt. Die steigende Flanke im folgenden Steuerungszyklus aktiviert die Drehzahlregelung mit der neuen, kleineren Solldrehzahl. Entsprechend dreht sich die Motorachse nach der Umschaltung des Signals *RapidSpeed_MDX1* auf *TRUE* wieder mit der größeren der beiden Solldrehzahlen [7].

Die durch eine steigende Flanke am Eingang *Execute* des Funktionsbausteins MC_AxisStop_MDX gestartete Bremsbewegung kann durch eine steigende Flanke am Eingang *Execute* des Funktionsbausteins MC_Stop_MDX abgebrochen werden. In Folge dessen wird eine Bremsbewegung mit der in den Parametern des Antriebsumrichters eingestellten Bremsrampe ausgeführt [8].

3.4 Tippbetrieb einer Motorachse ohne Geber

Aufgaben-beschreibung

In diesem Programmierbeispiel soll der Tippbetrieb mit zwei Geschwindigkeiten der Motorachse realisiert werden.

Dazu werden zwei digitale Eingänge des angesteuerten Antriebsumrichters MOVIDRIVE® MDX60B/61B für die Signale *Tippen Plus* bzw. *Tippen Minus* verwendet. Wenn an genau einem der beiden digitalen Eingänge die 24V DC Spannung anliegt, soll sich die Motorachse im Tippbetrieb drehen. Ansonsten muss der Antrieb abgebremst werden. Mit Hilfe eines weiteren digitalen Eingangs soll die Drehzahl der Motorachse zwischen zwei Werten umschaltbar sein.

Die Bedienung des Antriebs erfolgt vollständig durch Verwendung der digitalen Eingänge. Die Steuerung MOVI-PLC® wertet die Eingänge des Antriebsumrichters aus und steuert die Drehzahlregelung der Motorachse.

Programmierung

Unveränderte Schritte

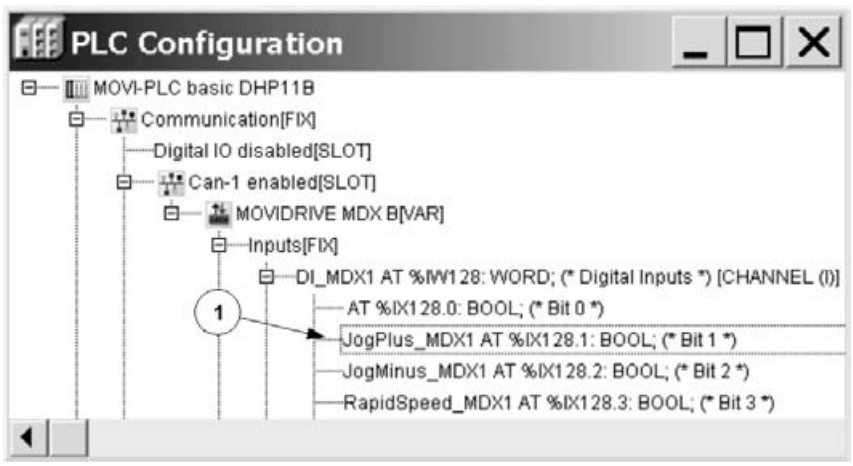
Programmieren Sie die Schritte 1, 3, 4 und 7 unverändert gemäß den vorhergehenden Programmierbeispielen "Positionieren einer Motorachse" und "Drehzahlregelung einer Motorachse mit Geber".

Hinweis:

Den Schritt 5 dürfen Sie nicht programmieren weil der Funktionsbaustein MC_Power_MDX für geberlose Motorachsen nicht angewendet wird.

Schritt 2

Steuerungskonfiguration erstellen


PLC Configuration
MOVI-PLC basic DHP11B
Communication[FIX]
Digital IO disabled[SLOT]
Can-1 enabled[SLOT]
MOVIDRIVE MDX B[VAR]
Inputs[FIX]
DI_MDX1 AT %IW128: WORD; (* Digital Inputs *) [CHANNEL (I)]
1
AT %IX128.0: BOOL; (* Bit 0 *)
JogPlus_MDX1 AT %IX128.1: BOOL; (* Bit 1 *)
JogMinus_MDX1 AT %IX128.2: BOOL; (* Bit 2 *)
RapidSpeed_MDX1 AT %IX128.3: BOOL; (* Bit 3 *)

20091AXX

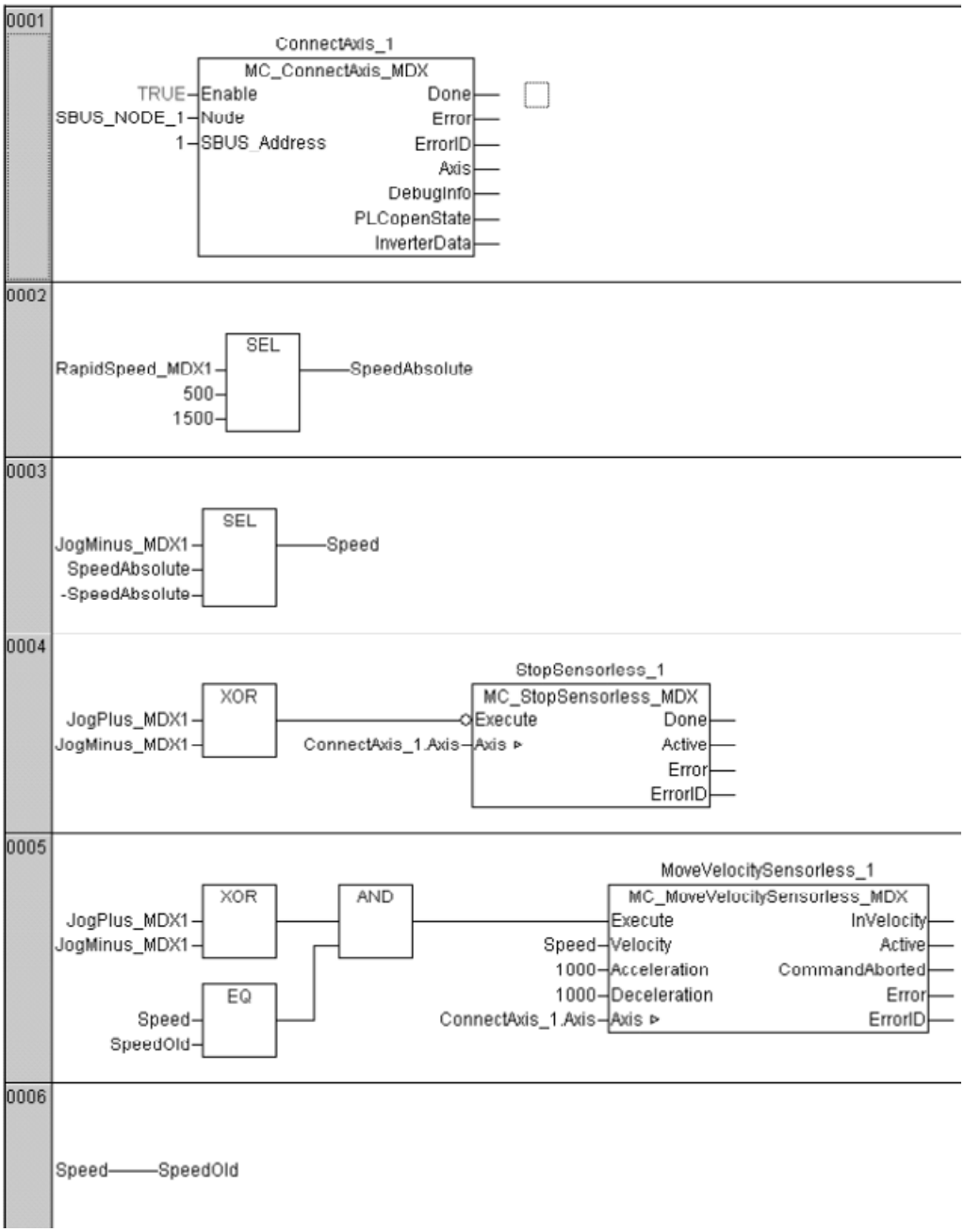
Ordnen Sie in der Steuerungskonfiguration die Bezeichner

- [JogPlus_MDX1]
- [JogMinus_MDX1]
- [RapidSpeed_MDX1]

den digitalen Eingängen des Antriebsumrichters MOVIDRIVE® MDX60B/61B gemäß der Abbildung zu [1].

Schritt 6

Programmierung des Tippbetriebs



20092AXX

Erstellen Sie gemäß der beschriebenen Vorgehensweise im den vorhergehenden Programmierbeispielen das in der Abbildung gezeigte Programm.

Schritt 8

Programm testen

Führen Sie das Programm aus, indem Sie die verwendeten digitalen Eingänge entsprechend schalten.

Warnung:

In Abhängigkeit von der Klemmenbelegung, des Antriebsumrichter-Zustandes und des Steuerungsprogramms kann sich die Motorachse nach dem Starten der Steuerung MOVI-PLC® sofort bewegen. Um Schäden zu vermeiden halten Sie ausreichend Abstand von allen bewegten Teilen!

Schalten Sie den Antriebsumrichter MOVIDRIVE® MDX60B/61B ein, wenn dies nicht bereits in Schritt 1 zum Einschalten der Steuerung MOVI-PLC® erfolgt ist.

Starten Sie die Steuerung MOVI-PLC®, indem Sie auf den Menüpunkt [Online] / [Start] ([Run]) klicken.

Deaktivieren Sie die Reglersperre, indem Sie am Eingang *DI00* "/ Reglersperre" die 24V DC Spannung anlegen.

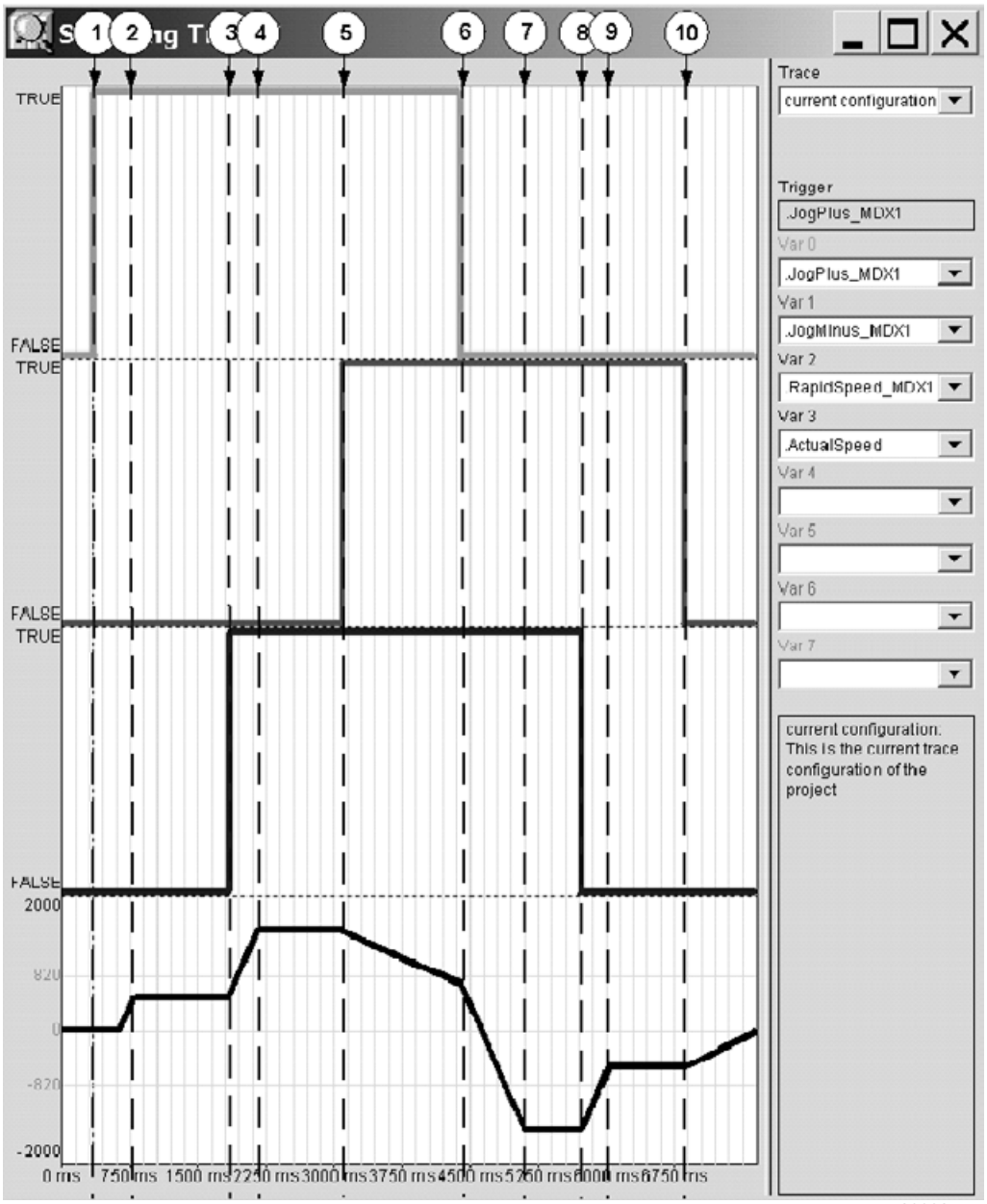
Starten Sie den Tippbetrieb der Motorachse, indem Sie an genau einem der beiden Eingänge *DI01* oder *DI02* des Antriebsumrichters die 24V DC Spannung anlegen.

Die Programmierung war erfolgreich, wenn

- sich die Motorachse bei angelegter 24V DC Spannung am Eingang *DI01* bzw. *DI02* in positiver (nach rechts) bzw. negativer Richtung (nach links) dreht,
- der Absolutwert der Motordrehzahl durch Wechsel zwischen 0V und 24V DC Spannung am Eingang *DI03* zwischen 500 1/min und 1000 1/min umschaltet und
- die Bewegung der Motorachse durch Anlegen der 24V DC Spannung an den beiden Eingängen *DI01* und *DI02* oder durch Wegnahme der Spannung von diesen beiden Eingängen abgebremst wird.

Weitere Details zum Verhalten der Steuerung MOVI-PLC® und des angeschlossenen Antriebsumrichters MOVIDRIVE® in diesem Beispiel entnehmen Sie bitte dem folgenden Abschnitt "Traceaufzeichnung".

Traceaufzeichnung



20093AXX

Bei einer steigenden Flanke des Signals *JogPlus_MDX1* startet die Motorachse die Drehzahlregelung mit der durch das Signal *RapidSpeed_MDX1* festgelegten Geschwindigkeit [1].

Beim Wechsel des Signals *RapidSpeed_MDX1* von *FALSE* auf *TRUE* erhöht die Motorachse die Geschwindigkeit auf den höheren der beiden vorgegebenen Werte [3].

Wenn die beiden Signale *JogPlus_MDX1* und *JogMinus_MDX1* gleichzeitig auf *TRUE* gesetzt werden, führt die XOR-Verknüpfung in Verbindung mit der Negation im Steuerungsprogramm zu einer steigenden Flanke am Eingang *Execute* des Funktionsbausteins MC_StopSensorless_MDX. Die Bremsbewegung wird gestarted [5].

Bei geberlosen Achsen lässt sich die durch die steigende Flanke am Eingang *Execute* des Funktionsbausteins MC_StopSensorless_MDX ausgelöste Bremsbewegung abbrechen. Dieser Abbruch erfolgt durch die Anforderung einer drehzahlgeregelten Bewegung durch die steigende Flanke am Eingang Execute des Funktionsbausteins

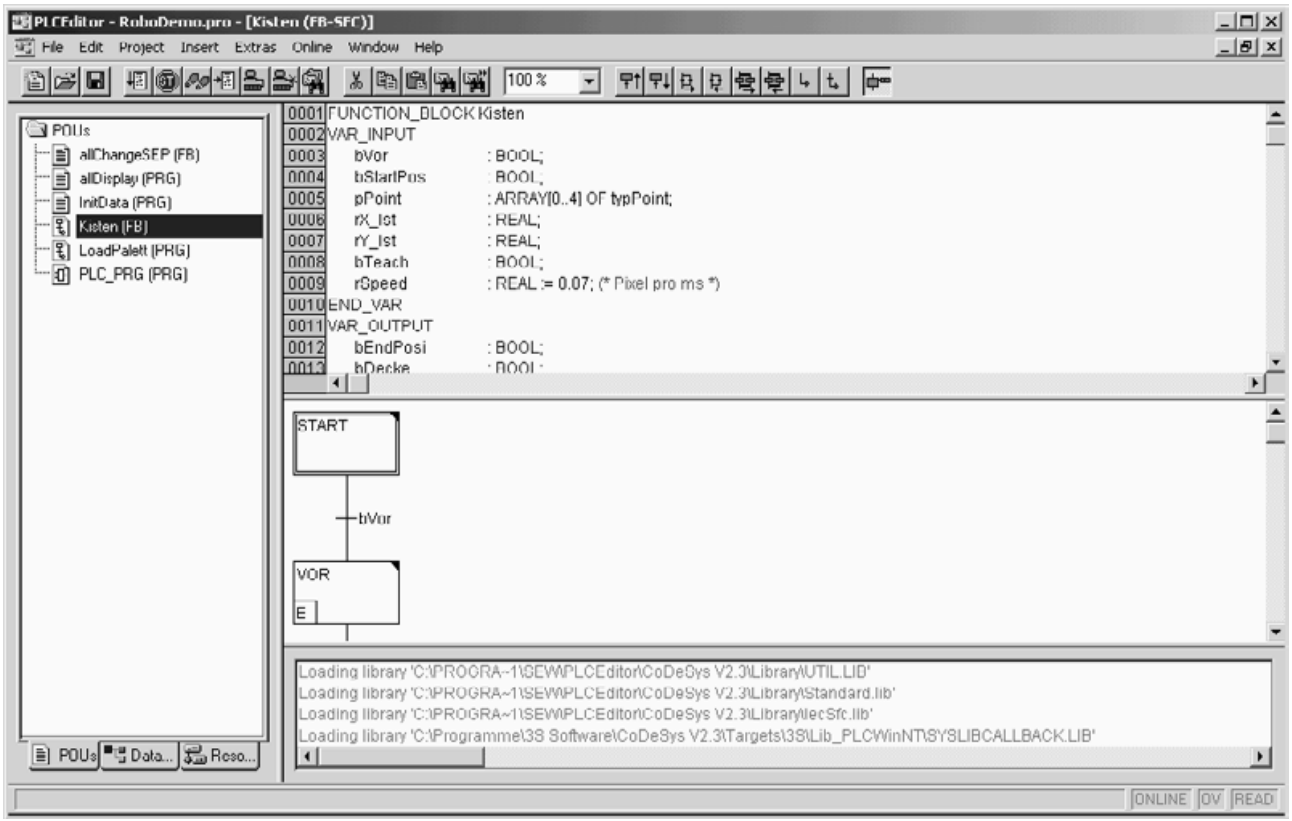
MC_MoveVelocitySensorless_MDX. [6]. In diesem Beispiel wird die erforderliche steigende Flanke durch die XOR-Verknüpfung im Netzwerk [0005] erzeugt, sobald das Signal *JogPlus_MDX1* von *TRUE* auf *FALSE* wechselt.

Beim erneuten Wechsel des Signals *RapidSpeed_MDX1* von *TRUE* auf *FALSE* wird die Motorachse auf die kleinere der beiden vorgegebenen Geschwindigkeiten abgebremst [8].

Das Zurücksetzen beider Signale *JogPlus_MDX1* und *JogMinus_MDX1* auf *FALSE* führt wie das zuvor beschriebene gleichzeitige Setzen der Signale auf *TRUE* zum Starten einer Bremsbewegung [10].

4 Komponenten des PLC-Editor

4.1 Das Hauptfenster



20254AEN

[1] Menüleiste
[2] Funktionsleiste
[3] Object Organizer
[4] Horizontale und vertikale Bildschirmteiler
[5] Arbeitsbereich
[6] Meldungsfenster
[7] Statusleiste

Menüleiste

Die Menüleiste enthält alle Menübefehle.

20255AEN

Die Menübefehle sind außerdem wählbar über

- Tastaturbefehle (zum Teil)
- die Funktionsleiste
- das Kontextmenü

Funktionsleiste

Die Schaltflächen der Funktionsleiste ermöglichen einen Schnellzugriff auf bestimmte Menübefehle. Der Inhalt der Funktionsleiste passt sich automatisch an das aktive Fenster an.

20256AXX

Wenn Sie den Mauszeiger kurze Zeit über ein Symbol in der Funktionsleiste halten, wird der Name des Symbols in einem **Tooltip** angezeigt.

Wenn Sie einen Menübefehl markiert aber noch nicht betätigt haben, erscheint eine kurze Beschreibung in der Statusleiste.

Eine Beschreibung aller Symbole finden Sie auf Seite 426 (→ Kap. 11, Anhang).

Die Anzeige der Funktionsleiste ist optional und wird wie folgt ein- und ausgeschaltet:

1. Führen Sie den Menübefehl [Projekt] / [Optionen] ([Project] / [Options]) aus.
2. Klicken Sie auf die Kategorie [Arbeitsbereich] ([Desktop]).
3. Um die Anzeige ein- bzw. auszuschalten, aktivieren bzw. deaktivieren Sie die Option [Funktionsleiste] ([Toolbar]).

Object Organizer

Der Object Organizer befindet sich an der linken Seite des Programmfensters und enthält drei Registerkarten.

Die Symbole der drei Registerkarten haben folgende Bedeutung:

Registerkarte	Objektkategorie
	Bausteine
	Datentypen
	Ressourcen

Tabelle 6: Registerkarten des Object Organizer

Zum Wechseln der Objektkategorie klicken Sie mit der Maus auf die Registerkarte oder benutzen Sie die linke bzw. rechte Pfeiltaste.

Zusätzliche Symbole vor oder hinter den Objekteinträgen kennzeichnen den Status hinsichtlich **Online Change**.

Sie können die Breite des Object Organizer ändern, indem Sie dessen rechten Rahmen mit der Maus verschieben.

Wie Sie mit den Objekten im Object Organizer arbeiten, erfahren Sie auf Seite 156 (→ Kap. 4.4, Objekte verwalten).

Bildschirmteiler

Der Bildschirmteiler ist die Grenze zwischen zwei sich nicht überlappenden Fenstern.

Im PLC-Editor gibt es Bildschirmteiler zwischen folgenden Fenstern:

- **Object Organizer** und Arbeitsbereich
- Deklarationsteil (Schnittstelle) und Anweisungsteil (Implementierung) von Bausteinen
- Arbeitsbereich und Meldungsfenster

Wenn Sie den Mauszeiger über einen Bildschirmteiler führen, ändert sich der Mauszeiger in einen Doppelpfeil. Klicken Sie jetzt auf den Bildschirmteiler und verschieben Sie ihn bei gedrückter Maustaste.

Der Bildschirmteiler bleibt stets an seiner absoluten Position, auch wenn Sie die Fenstergröße verändern. Wenn ein Bildschirmteiler verschwunden zu sein scheint, verbrei-

tern Sie das Programmfenster bis er wieder angezeigt wird.

Arbeitsbereich

Der Arbeitsbereich befindet sich auf der rechten Seite des Hauptfensters. Alle Editorfenster und die Bibliotheksverwaltung werden im Arbeitsbereich geöffnet.

In der Titelleiste der Editorfenster erscheint der Objektname. Bei Bausteinen stehen in einer Klammer zusätzlich je ein Kürzel für den Bausteintyp und die verwendete Programmiersprache.

Unter dem Menüpunkt [Fenster] ([Window]) befinden sich alle Menübefehle zur Fensterverwaltung.

Meldungsfenster

Das Meldungsfenster befindet sich unterhalb des Arbeitsbereichs.

Das Meldungsfenster enthält alle Meldungen aus dem letzten Übersetzungs-, Überprüfungs- oder Vergleichsvorgang. Suchergebnisse und die Querverweisliste werden hier ebenfalls ausgegeben.

Wenn Sie im Meldungsfenster auf eine Meldung doppelklicken oder die <Eingabe>-Taste drücken, wird der Editor mit dem dazugehörigen Objekt geöffnet und die Zeile des Objekts markiert, die die Meldung verursacht hat.

Mit den Menübefehlen [Bearbeiten] / [Nächster Fehler] ([Edit] / [Next Error]) und [Bearbeiten] / [Vorheriger Fehler] ([Edit] / [Previous Error]) wechseln Sie zwischen den Fehlermeldungen.

Die Anzeige des Meldungsfensters ist optional und wird über den Menübefehl [Fenster] / [Meldungen] ([Window] / [Messages]) oder den Tastaturbefehl <Umschalt>+<Esc> ein- und ausgeschaltet.

Statusleiste

Die Statusleiste befindet sich ganz unten im Hauptfenster und zeigt Informationen zum aktuellen Projekt und zu den Menübefehlen an.

Wenn ein Eintrag rechts in der Statusleiste zutrifft, wird er in schwarzer Schrift dargestellt, ansonsten in grauer Schrift.

Wenn Sie im Online-Modus arbeiten, erscheint der Begriff **ONLINE** in schwarzer Schrift. Wenn Sie im Offline-Modus arbeiten, erscheint der Begriff **ONLINE** in grauer Schrift.

Im Online-Modus werden außerdem folgende Informationen angezeigt:

- **SIM**: Simulation läuft.
- **LÄUFT**: Programm wird abgearbeitet.
- **BP**: Breakpoint gesetzt.
- **FORCE**: Forcing von Variablen.

Bei Texteditoren wird die Zeilen- und Spaltennummer der aktuellen Cursorposition angegeben im Format Z.: 2, Sp.: 7.

Wenn Sie im Überschreibmodus arbeiten, wird **ÜB** schwarz angezeigt. Durch Drücken der <Einfg>-Taste wechseln Sie zwischen Überschreib- und Einfügemodus.

Die Anzeige der Statusleiste ist optional und wird wie folgt ein- und ausgeschaltet:

1. Wählen Sie den Menübefehl [Projekt] / [Optionen] ([Project] / [Options]).
2. Klicken Sie auf die Kategorie [Arbeitsbereich] ([Desktop]).
3. Um die Anzeige der Statusleiste ein- bzw. auszuschalten, aktivieren bzw. deaktivieren Sie die Option [Statusleiste] ([Statusbar]).

Kontextmenü

Wenn Sie auf ein Objekt oder einen Editor mit der rechten Maustaste klicken (Tastaturbefehl: <Umschalt>+<F10>), erscheint das Kontextmenü.

Das Kontextmenü bietet einen Schnellzugriff auf die wichtigsten Menübefehle des Objekts oder Editors.

4.2 Projektoptionen

4.2.1 Menübefehl [Projekt] / [Optionen] ([Projects] / [Options])

Mit dem Menübefehl [Projekt] / [Optionen] können Sie z. B. die Ansicht des PLC-Editor-Hauptfensters konfigurieren. Soweit nicht anders vermerkt, werden die Einstellungen in der Datei `PLCEditor.ini` gespeichert und beim nächsten Start des PLC-Editors wiederhergestellt.

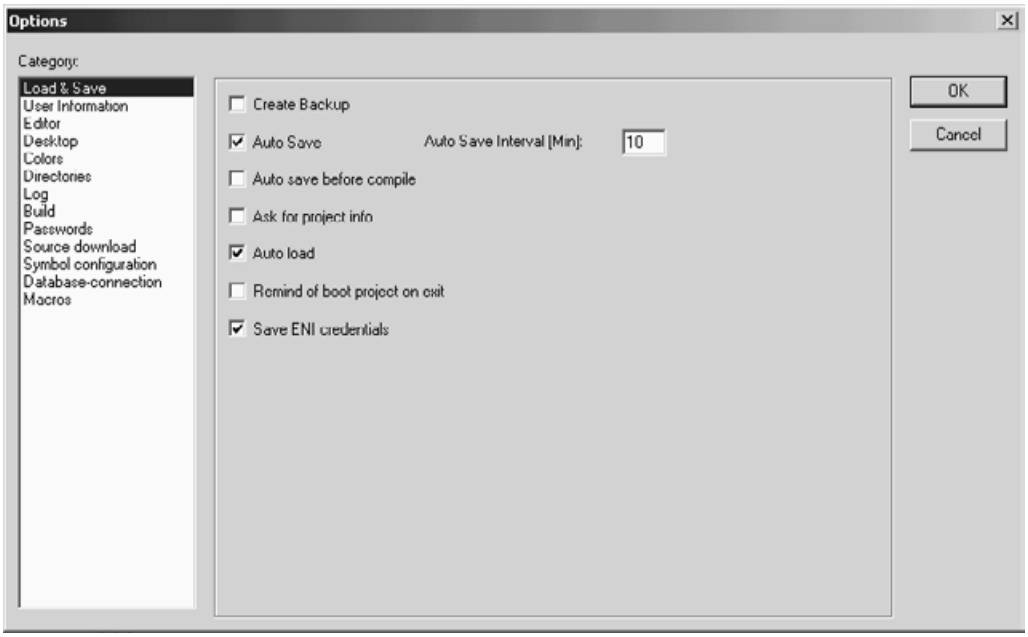
Ein Abbild der für das Projekt eingestellten Projektoptionen wird im **Object Organizer**, Registerkarte [Ressourcen] ([Resources]) im Knoten [Arbeitsbereich] ([Workspace]) abgelegt.

Nach dem Ausführen des Menübefehls [Projekt] / [Optionen] wird der Dialog [Optionen] ([Options]) geöffnet. Der Dialog ist in Kategorien aufgeteilt. Wählen Sie links im Dialog die gewünschte Kategorie durch einen Mausklick oder mit den Pfeiltasten und verändern Sie rechts die Einstellungen der Kategorie.

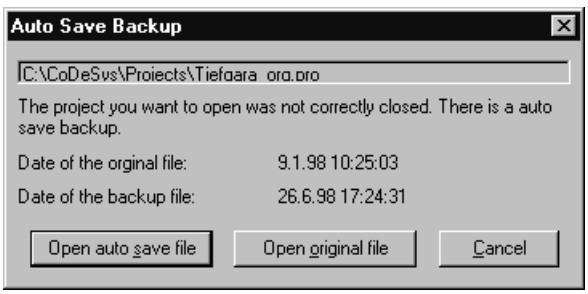
Folgende Kategorien sind enthalten:

Kategorie	
[Load & Save]	[Laden & Speichern]
[User information]	[Benutzerinformation]
[Editor]	[Editor]
[Desktop]	[Arbeitsbereich]
[Colors]	[Farben]
[Directories]	[Verzeichnisse]
[Logbuch]	[Log]
[Build]	[Übersetzungsoptionen]
[Passwords]	[Kennworte]
[Source download]	[Sourcedownload]
[Symbol configuration]	[Symbolkonfiguration]
[Database-connection]	[Projektdatenbank]
[Macros]	[Makros]

4.2.2 [Laden & Speichern] ([Load & Save])

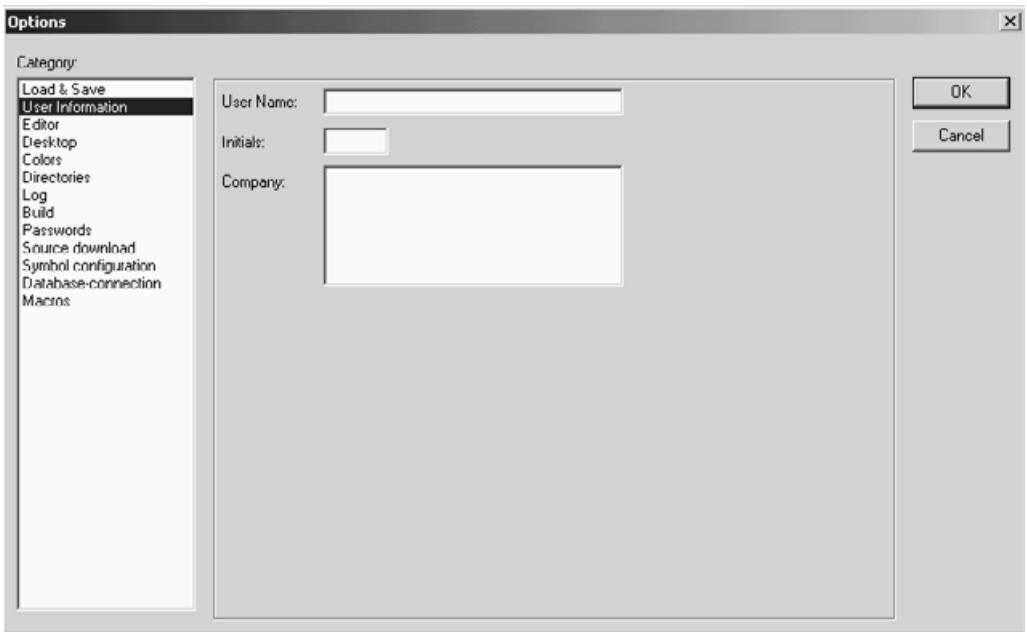


20257AEN

[Sicherheitskopie erstellen]	[Create Backup]	Der PLC-Editor speichert die geöffnete Datei bei jedem Menübefehl [Datei] / [Speichern] ([File] / [Save]) zusätzlich in eine separate Sicherungsdatei mit der Erweiterung `.bak`. Diese Datei bleibt im Gegensatz zu der `*.asd`-Sicherungsdatei (siehe Option [Automatisch sichern]) auch nach Beenden des Projekts erhalten. Sie können daraus also stets die Version vor der letzten Speicherung wieder herstellen.
[Automatisch sichern], [Sicherungsintervall [Min.]]	[Auto Save], [Auto Save Interval [Min.]]	Das geöffnete Projekt wird in dem eingestellten Zeitintervall [Sicherungsintervall [Min.]] in eine temporäre Datei mit der Erweiterung `.asd` gespeichert. Diese Datei wird beim normalen Beenden des PLC-Editors gelöscht. Wenn der PLC-Editor nicht korrekt beendet wird – z. B. bei einem Stromausfall – wird diese Datei nicht gelöscht. Wenn Sie ein Projekt nach einem solchen Fehlerfall erneut öffnen, wird der Dialog [Automatische Sicherung] ([Auto Save Backup]) geöffnet.  In diesem Dialog können Sie entscheiden, ob Sie die Originaldatei oder die Sicherungsdatei öffnen möchten.
[Automatisch sichern vor Übersetzen]	[Auto save before compile]	Das Projekt wird vor jedem Übersetzungslauf gespeichert. Dabei wird eine Datei mit der Erweiterung `.asd` angelegt, die sich verhält wie bei der Option [Automatisch sichern] beschrieben.
[Projektinformation verlangen]	[Ask for project info]	Beim Abspeichern eines neuen Projekts oder beim Abspeichern eines Projekts unter einem neuen Namen wird der Dialog [Projektinformation] ([Project Information]) aufgerufen. Die Projektinformationen können Sie jederzeit mit dem Menübefehl [Projekt] / [Projektinformation] ([Project] / [Project Info]) einsehen und bearbeiten.
[Automatisch laden]	[Auto load]	Beim nächsten Start des PLC-Editor wird das zuletzt geöffnete Projekt automatisch geladen. Das Laden eines Projekts kann beim Start des PLC-Editor auch durch Angabe eines Projekts in der Befehlszeile erfolgen.
[An Bootprojekt erinnern vor Beenden]	[Remind of boot project on exit]	Wenn seit dem Erzeugen eines Bootprojekts das Projekt in modifizierter Form auf die Steuerung geladen wurde, ohne ein neues Bootprojekt zu erzeugen, werden Sie beim Verlassen des Projekts darauf aufmerksam gemacht.

[Zugangsdaten für Projektdatenbank speichern]	[Save ENI credentials]	Falls Sie für die ENI-Datenbank einen Benutzernamen und ein Passwort angegeben haben, werden diese gespeichert. Die unter dem Menübefehl [Datei] / [Öffnen] / [Projekt aus Projektdatenbank öffnen] ([File] / [Open] / [Open project from PLC]) eingegebenen Zugangsdaten (Benutzername und Passwort) werden in der Datei `PLCEditor.ini` gespeichert.

4.2.3 [Benutzerinformation] ([User information])

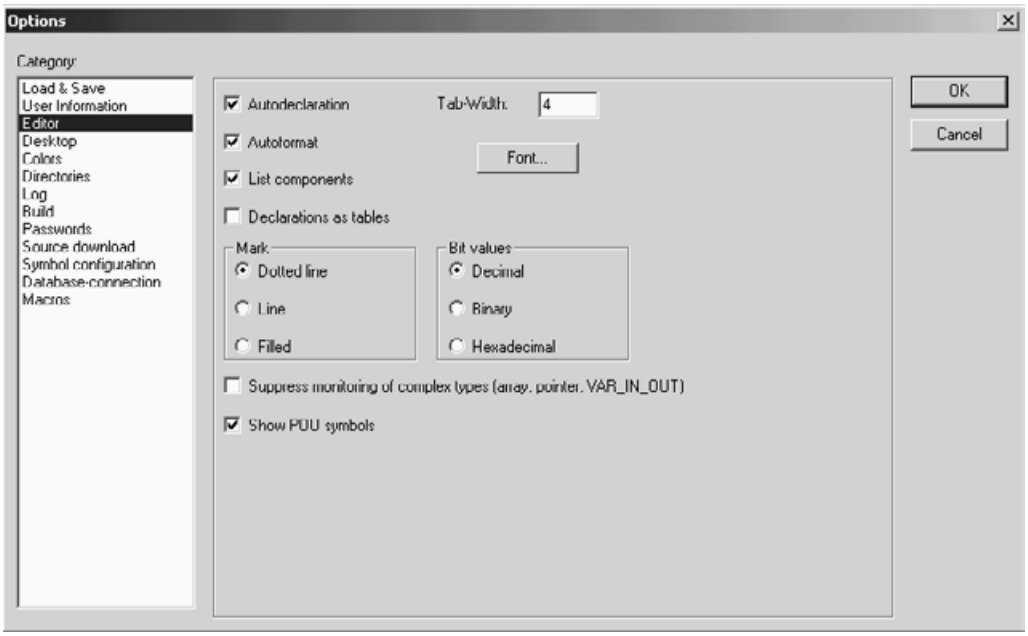


20258AEN

[Benutzername]	[User Name]	Geben Sie hier Ihren Namen ein.
[Initialen]	[Initials]	Geben Sie hier Ihre Initialen ein.
[Firma]	[Company]	Hier können Sie Angaben zu Ihrer Firma machen.

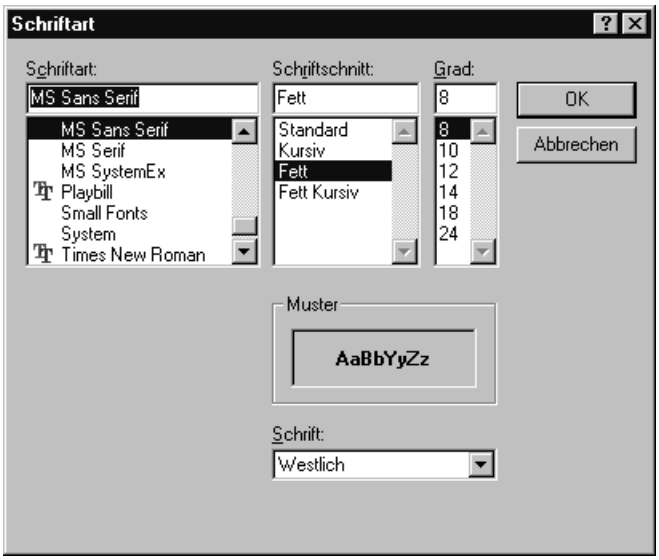
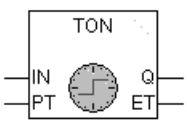
Die Angaben in dieser Kategorie werden für weitere Projekte automatisch übernommen.

4.2.4 [Editor] ([Editor])

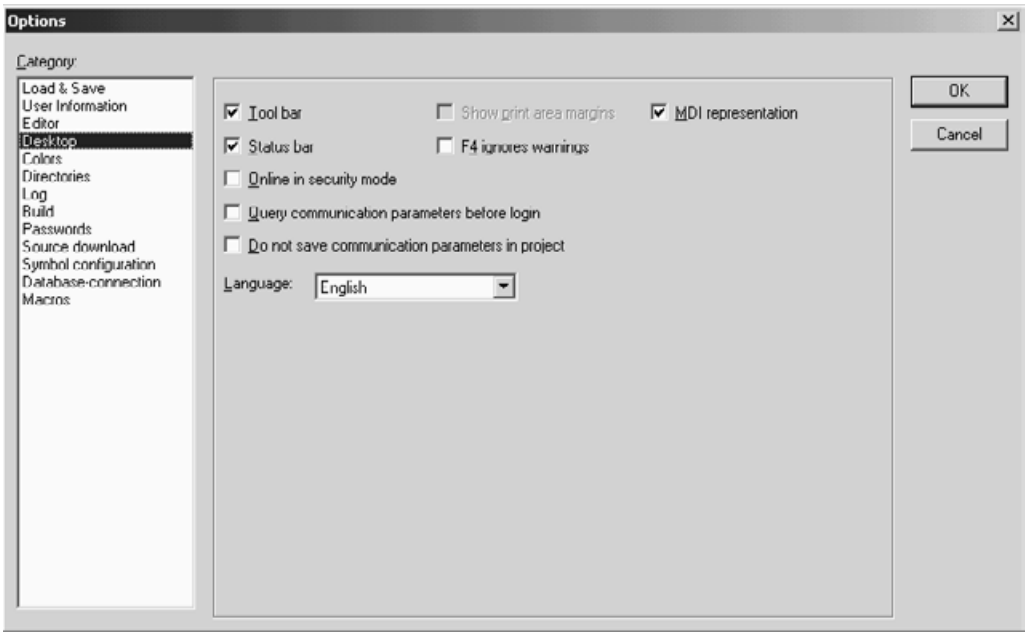


20259AEN

[Automatisch deklarieren]	[Autodeclaration]	In allen Editoren erscheint nach Eingabe einer noch nicht deklarierten Variablen der Dialog [Variablendeklaration] ([Declare Variable]), mit dessen Hilfe diese Variable deklariert werden kann.
[Automatisch formatieren]	[Autoformat]	Der PLC-Editor führt eine automatische Formatierung im Anweisungslisten- und im Deklarationseditor durch. Wenn eine Zeile verlassen wird, werden folgende Formatierungen durchgeführt: • Operatoren und Schlüsselwörter, die in Kleinbuchstaben geschrieben sind, werden in Großbuchstaben dargestellt. • Tabulatoren werden eingefügt, so dass sich eine einheitliche Spaltenaufteilung ergibt.
[Komponenten auflisten]	[List components]	Wenn diese Option aktiviert ist, steht die **Intellisense**-Funktion im PLC-Editor zur Verfügung. Wenn Sie dann an Stellen, an denen ein Bezeichner eingegeben werden soll, nur einen Punkt eingeben, erhalten Sie eine Auswahlliste aller im Projekt verfügbaren globalen Variablen. Wenn Sie den Namen einer Funktionsblockinstanz, gefolgt von einem Punkt, eingeben, erhalten Sie eine Auswahlliste der Ein-und Ausgänge des instanzierten Funktionsblocks. Die Intellisense-Funktion gibt es in folgenden Programmteilen: • Editoren • Watch- und Rezepturverwalter • Visualisierung • Tracekonfiguration
[Deklarationen als Tabellen]	[Declarations as tables]	Variablen können außer mit dem Deklarationseditor auch mit einem Editor in Tabellenform editiert werden. Diese Tabelle ist wie ein Karteikasten mit Registerkarten geordnet. Es gibt die Registerkarten `VAR`, `VAR_INPUT`, `VAR_OUTPUT`, `CONSTANT`, `RETAIN` und `INFO`. Für jede Variablendeklaration gibt es folgende Felder: • [Name] ([Name]) • [Adresse] ([Address]) • [Typ] ([Type]) • [Initial] ([Initial]) • [Kommentar] ([Comment])
[Tab-Breite]	[Tab-Width]	Im Eingabefeld [Tab-Breite] können Sie eingeben, wie breit ein Tabulator in den Editoren dargestellt wird. Voreingestellt ist eine Breite von vier Zeichen, wobei die Zeichenbreite wiederum von der eingestellten Schriftart abhängt.

[Schrift]	[Font]	Nach dem Klicken auf die Schaltfläche [Font] wird der Dialog [Schriftart] ([Font]) geöffnet.  Die hier eingestellten Schriftmerkmale (Schriftart, Schriftschnitt und Grad) werden in allen Editoren verwendet. Die Größe der Schrift ist die Grundeinheit für alle Zeichnungsoperationen. Die Einstellung einer anderen Schriftgröße verändert sowohl die Ausgabe als auch den Ausdruck bei jedem Editor.
[Markierung]	[Mark]	Die Einstellung [Markierung] regelt die Darstellung des Markierungsrechtecks in den grafischen Editoren. Zur Auswahl stehen: • [Gepunktet] ([Dotted line]) • [Linie] ([Line]) • [Ausgefüllt] ([Filled])
[Bitwerte]	[Bit values]	Sie können bestimmen, wie binäre Datentypen (`BYTE`, `WORD`, `DWORD`) beim Monitoring dargestellt werden sollen: • [Dezimal] ([Decimal]) • [Binär] ([Binary]) • [Hexadezimal] ([Hexadecimal])
[Monitoring von komplexen Typen (Array, Pointer, VAR_IN_OUT) unterbinden]	[Suppress monitoring of complex type (array, pointer, VAR_IN_OUT)]	Wenn die Option [Monitoring von komplexen Typen (Array, Pointer, VAR_IN_OUT) unterbinden] aktiviert ist, werden komplexe Datentypen wie `Array`, `Pointer`, `VAR_IN_OUT` nicht im Monitorfenster des Online-Modus dargestellt.
[Bausteinsymbole anzeigen]	[Show POU symbols]	Wenn die Option [Bausteinsymbole anzeigen] aktiviert ist, werden in den Bausteinboxen Symbole dargestellt, sofern diese als Bitmaps im Bibliotheksverzeichnis vorliegen. Der Name der Bitmap-Datei muss sich aus dem Bausteinnamen und der Erweiterung `.bmp` zusammensetzen. Beispiel: Für den Baustein `TON` ist das Symbol in der Datei `TON.bmp` enthalten: 

4.2.5 [Arbeitsbereich] ([Desktop])

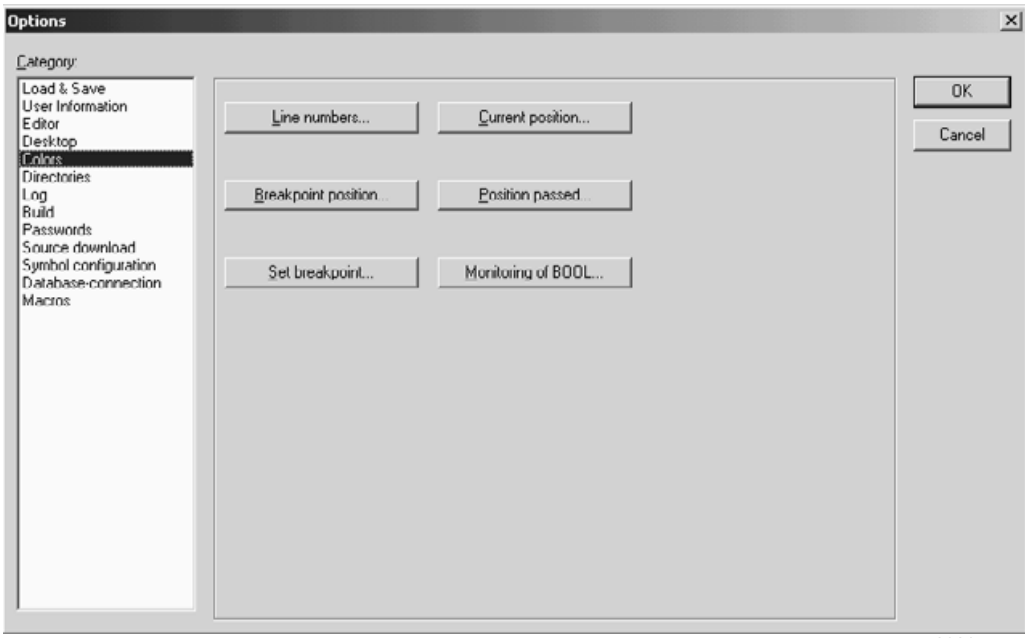


20260AEN

[Funktionsleiste]	[Tool bar]	Die Funktionsleiste mit den Schaltflächen für den Schnellzugriff auf Menübefehle wird unterhalb der Menüleiste sichtbar.
[Statusleiste]	[Status bar]	Die Statusleiste wird am unteren Rand des Hauptfensters sichtbar.
[Online-Betrieb im Sicherheitsmodus]	[Online in security mode]	Im Online-Modus erscheint bei bestimmten Befehlen ein Dialog mit der Sicherheitsabfrage, ob der Befehl wirklich ausgeführt werden soll. Die Sicherheitsabfrage erscheint bei folgenden Befehlen: • Start • Stop • Reset • Breakpoint an • Einzelzyklus • Werte schreiben • Werte "forcen" • Forcen aufheben Abhängig vom Laufzeitsystem erscheint beim Laden eines Projekts auf die Steuerung ein erweiterter Dialog. Der Dialog zeigt zusätzlich die Projektinformationen eines gegebenenfalls bereits auf der Steuerung vorhandenen sowie des neu zu ladenden Projekts an. Diese Projektinformationen erscheinen dann ebenfalls beim Erzeugen eines Bootprojekts, wenn bereits ein solches auf der Steuerung vorliegt. Die Option wird mit dem Projekt gespeichert.
[Kommunikationseinstellungen vor Login abfragen]	[Query communication parameters before login]	Nach dem Ausführen des Menübefehls [Online] / [Einloggen] ([Online] / [Login]) erscheint zunächst der Dialog [Kommunikationsparameter] ([Communication Parameters]). Erst nachdem dieser mit [OK] geschlossen wurde, wird in den Online-Modus gewechselt.
[Kommunikationseinstellungen nicht im Projekt speichern]	[Do not save communication parameters in project]	Die Einstellungen unter dem Menübefehl [Online] / [Kommunikationsparameter] ([Online] / [Communication Parameters]) werden nicht mit dem Projekt gespeichert.
[Druckbereiche anzeigen]	[Show print area margins]	In jedem Editorfenster werden die Begrenzungen des aktuell eingestellten Druckbereichs durch rot gestrichelte Linien markiert. Die Größe des Druckbereichs hängt von den Druckereigenschaften (Papiergröße, Ausrichtung) und der Größe des Arbeitsbereichs der eingestellten Druckvorlage ab.
[F4 ignoriert Warnungen]	[F4 ignores warnings]	Wenn Sie nach einem Übersetzungslauf im Meldungsfenster die <F4>-Taste drücken, springt der Fokus normalerweise zur Zeile mit der nächsten Fehlermeldung oder Warnung. Bei aktivierter Option [F4 ignoriert Warnungen] werden Warnungen ignoriert.

[MDI-Darstellung]	[MDI representation]	Standardmäßig ist die Option [MDI-Darstellung] (MDI: Multiple Document Interface) aktiviert und damit das gleichzeitige Öffnen mehrerer Objekte (Fenster) möglich. Wenn Sie diese Option deaktivieren, läuft das Projekt im SDI-Modus (SDI: Single Document Interface). Im SDI-Modus kann jeweils nur ein Fenster im Arbeitsbereich geöffnet sein, das dann im Vollbildmodus dargestellt wird. Ausnahme: Die Aktion eines Programms wird auch im SDI-Modus gleichzeitig mit dem Programm dargestellt.
[Sprache]	[Language]	Mit der Einstellung [Sprache] legen Sie fest, in welcher Sprache die Menü- und Dialogtexte sowie die Online-Hilfe erscheinen.

4.2.6 [Farben] ([Colors])

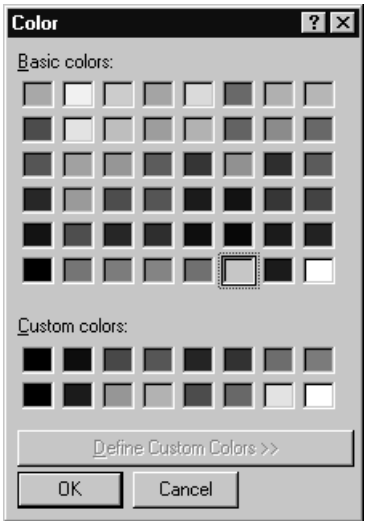


20261AEN

In der Kategorie [Farben] ändern Sie die Farbeinstellungen des PLC-Editor.

Eintrag		Voreingestellte Farbe
[Zeilennummern]	[Line numbers]	hellgrau
[Breakpoint Position]	[Breakpoint position]	dunkelgrau
[Gesetzter Breakpoint]	[Set breakpoint]	hellblau
[Aktuelle Position]	[Current position]	rot
[Durchlaufene Position]	[Position passed]	grün
[Monitoring von BOOL]	[Monitoring of BOOL]	blau

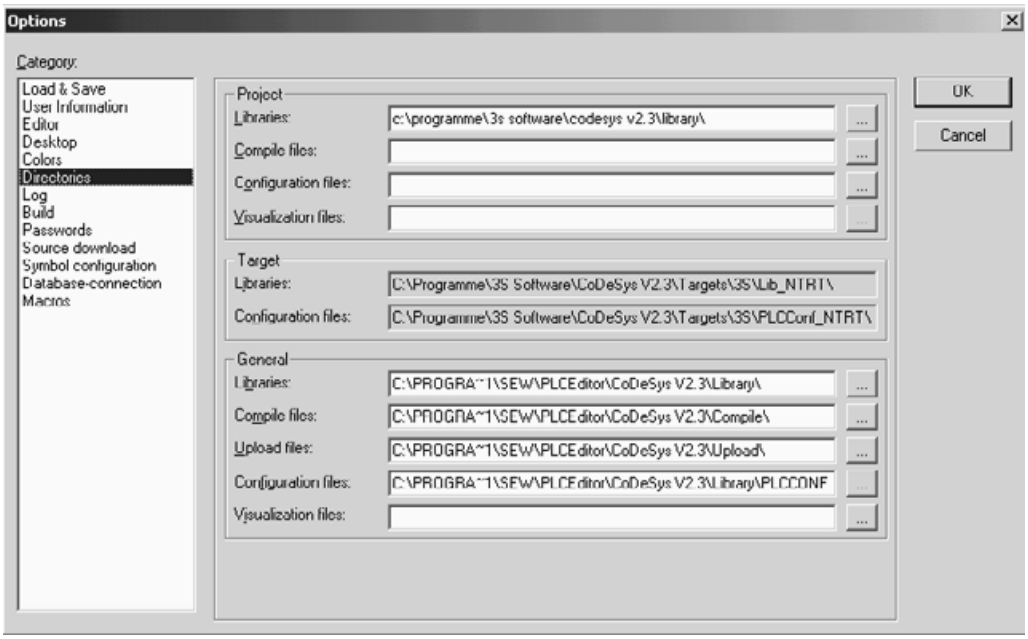
Wenn Sie auf eine dieser Schaltflächen klicken, erscheint jeweils folgender Dialog:



20272AEN

Wählen Sie die gewünschte Farbe aus und bestätigen Sie mit [OK] ([OK]).

4.2.7 [Verzeichnisse] ([Directories])



20262AEN

In den Bereichen [Projekt] und [Allgemein] ([Project] und [General]) geben Sie die Verzeichnispfade ein, die der PLC-Editor nach Bibliotheken, Steuerungskonfigurationsdateien und Visualisierungsdateien (Bitmaps, XML-Dateien für dynamische Texte etc.) durchsuchen bzw. für die Ablage von Datei für die Übersetzung und den Sourcedownload verwenden soll.

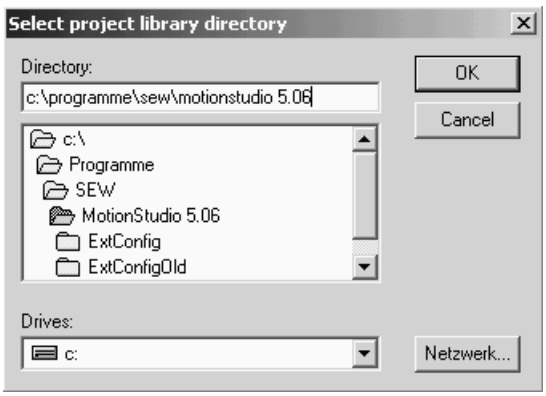
Hinweis:

Übersetzungsdateien sind beispielsweise Dateien mit den Endungen `*.map` und `*.list`, nicht jedoch Symboldateien! Letztere werden im Projektverzeichnis gespeichert.

Für Bibliotheks- und Konfigurationsdateien können Sie jeweils mehrere Pfade, getrennt durch ein Semikolon (;), eingeben.

Die Verzeichnispfade können Sie auf zwei Arten festlegen:

- Geben Sie die Verzeichnispfade direkt in die Eingabefelder ein.
- Klicken Sie auf die Schaltfläche [...]. Der Dialog [Bibliotheksverzeichnis für Projekt auswählen] ([Select project library directory]) wird geöffnet. Wählen Sie das gewünschte Verzeichnis aus und bestätigen Sie mit [OK] ([OK]).



20274AEN

Hinweis:

Bei Verzeichnispfaden für Bibliotheken sind Pfadangaben relativ zum aktuellen Projektverzeichnis möglich, indem ein Punkt (.) vorangestellt wird.

Beispiel: Das aktuelle Projekt liegt im Verzeichnis `C:\Programme\projects`. Wenn Sie für das Bibliotheksverzeichnis `.\libs` angeben, werden die Bibliotheken im Verzeichnis `C:\Programme\projects\libs` gesucht.

Weitere Informationen zu Bibliothekspfaden finden Sie auf Seite 286 (→ Kap. 6.4.5, [Einfügen] / [Weitere Bibliothek]).

Hinweis:

Verwenden Sie in den Verzeichnispfaden keine Leerzeichen und Sonderzeichen mit Ausnahme des Unterstrichs (_).

Die Angaben im Bereich [Projekt] ([Project]) werden mit dem Projekt gespeichert.

Die Angaben im Bereich [Allgemein] ([General]) werden in die `*.ini`-Datei des Programmiersystems geschrieben und gelten somit für alle Projekte.

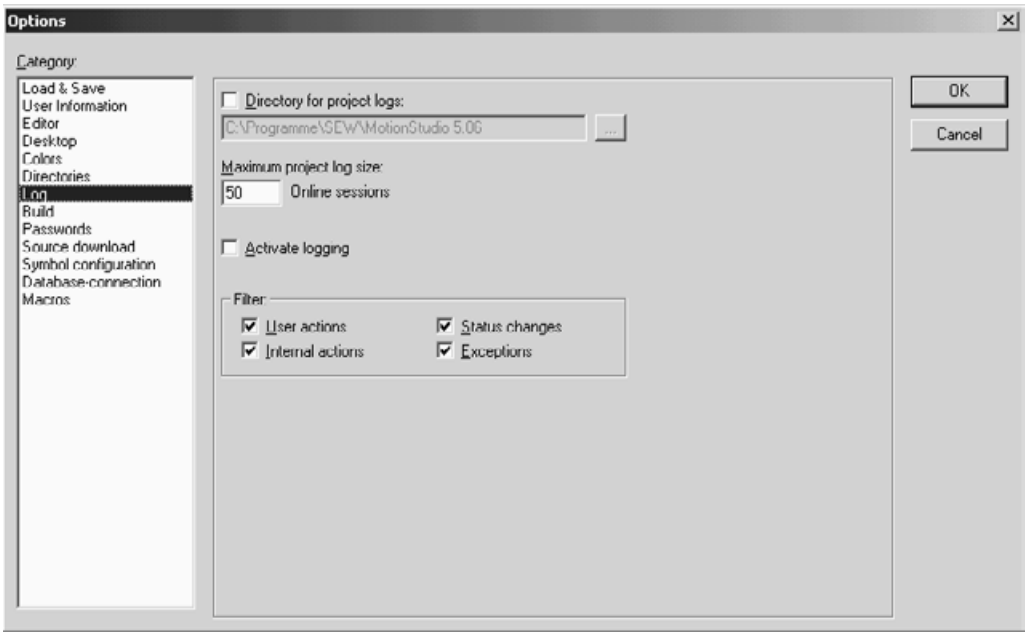
Im Bereich [Zielsystem] ([Target]) werden die Verzeichnisse für Bibliotheken und Konfigurationsdateien dargestellt, die im Zielsystem eingestellt sind, z. B. durch die Angaben in der Target-Datei. Diese Felder sind nicht editierbar, aber ein Eintrag kann mit der Maus markiert und kopiert werden (Kontextmenü über rechte Maustaste).

Der PLC-Editor durchsucht die Bereiche generell in folgender Reihenfolge:

- [Projekt]
- [Zielsystem]
- [Allgemein]

Wenn gleichnamige Dateien vorliegen, wird die zuerst gefundene Datei verwendet.

4.2.8 [Logbuch] ([Log])

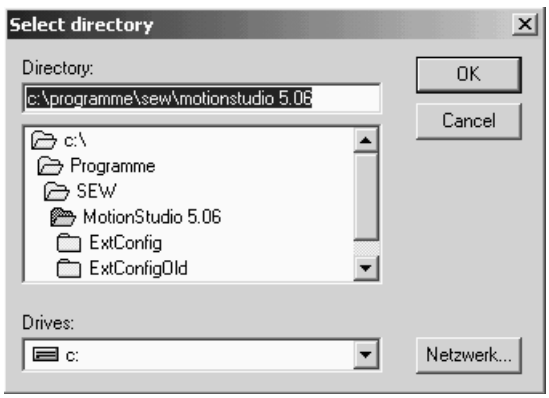


20263AEN

In der Kategorie [Logbuch] können Sie eine Datei konfigurieren, die als Projekt-Logbuch alle Benutzeraktionen und internen Vorgänge während des Online-Modus chronologisch aufzeichnet.

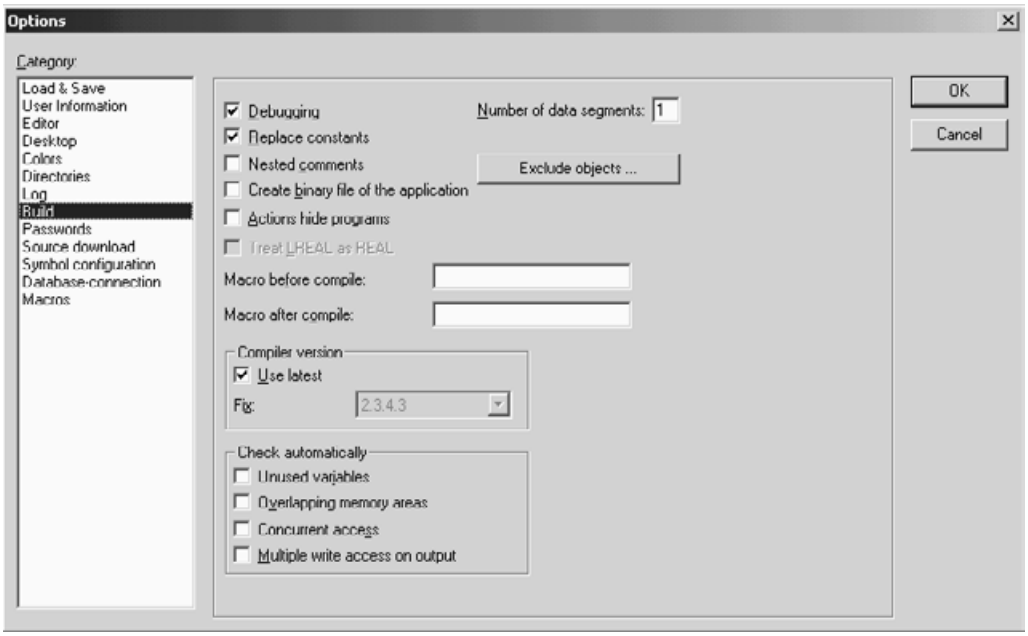
Wenn Sie ein bestehendes Projekt öffnen, für das noch kein Logbuch generiert wurde, erscheint ein Dialog mit dem Hinweis, dass nun ein Logbuch angelegt wird. Dieses Logbuch erhält mit dem nächsten Login-Vorgang die ersten Einträge.

Das Logbuch öffnen Sie mit dem Menübefehl [Fenster] / [Logbuch] ([Window] / [Log]).

[Verzeichnis für Projekt-Logbücher]	[Directory for project logs]	Standardmäßig wird das Logbuch (Dateiname: <Projektname>.log) als Binärdatei im Projektverzeichnis abgelegt. Wenn Sie ein anderes Dateiverzeichnis verwenden möchten, aktivieren Sie die Option [Verzeichnis für Projekt-Logbücher] und geben Sie den neuen Verzeichnispfad in das darunter liegende Feld ein. Alternativ können Sie über die Schaltfläche [...] den Dialog [Verzeichnis auswählen] ([Select directory]) öffnen und dort ein Verzeichnis auswählen. 
[Maximale Größe eines Projekt-Logbuchs]	[Maximum project log size]	Mit der Einstellung [Maximale Größe eines Projekt-Logbuchs] legen Sie die maximale Anzahl der Online-Sitzungen fest, die im Logbuch dokumentiert werden. Nach Erreichen der maximalen Anzahl wird jeweils der älteste Eintrag zugunsten des neuen gelöscht.
[Logbuch aktivieren]	[Activate logging]	Mit der Option [Logbuch aktivieren] legen Sie fest, ob Einträge in das Logbuch vorgenommen werden.

[Filter]	[Filter]	Im Bereich [Filter] stellen Sie ein, welche Vorgänge im Logbuch protokolliert werden: • [Benutzeraktionen] ([User actions]) • [Interne Aktionen] ([Internal actions]) • [Statusänderungen] ([Status changes]) • [Exceptions] ([Exceptions])

4.2.9 [Übersetzungsoptionen] ([Build])



20264AEN

Alle Einstellungen der Kategorie [Übersetzungsoptionen] werden mit dem Projekt gespeichert.

[Debugging]	[Debugging]	Die Option [Debugging] ist zielsystemabhängig aktivbar bzw. voreingestellt. Wenn die Option aktiviert ist, wird zusätzlich ein Debugging-Code erzeugt. Der Code kann dadurch deutlich umfangreicher werden. Außerdem wird das Projekt langsamer abgearbeitet. Der Debugging-Code ist notwendig, um die vom PLC-Editor angebotenen Debugging-Funktionen benutzen zu können.
[Konstanten ersetzen]	[Replace constants]	Wenn die Option [Konstanten ersetzen] aktiviert ist, wird für jede Konstante direkt deren Wert geladen. Die Konstanten werden im Online-Modus grün dargestellt. Forcing, Schreiben und Monitoring sind dann bei Konstanten nicht mehr möglich. Wenn die Option deaktiviert ist, wird der Wert über Variablenzugriff auf einen Speicherplatz geladen. Dies ermöglicht das Schreiben des Variablenwerts, bedeutet aber eine längere Bearbeitungszeit.

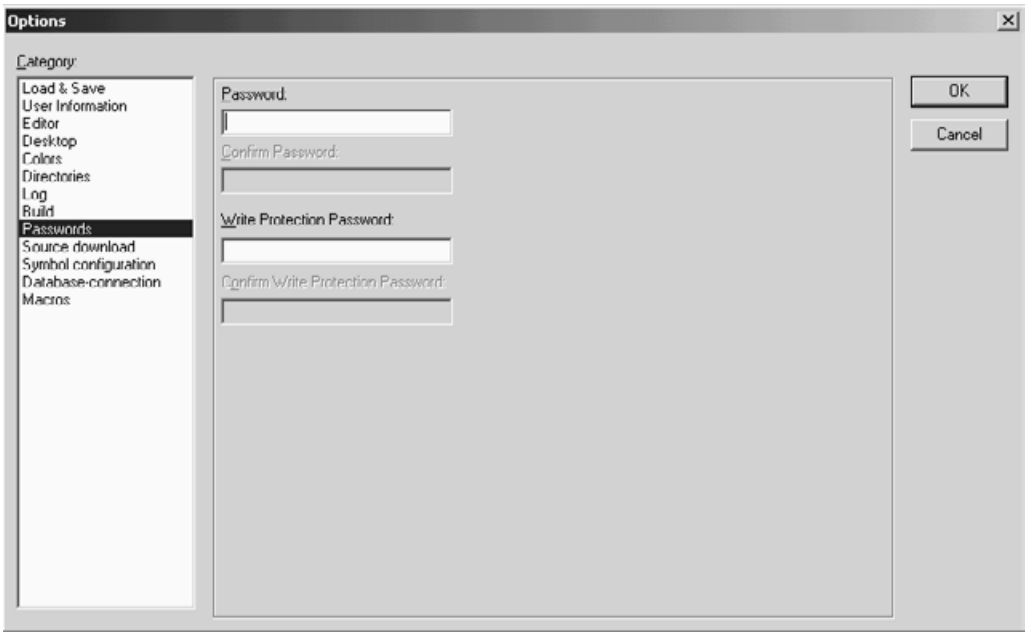
[Verschachtelte Kommentare]	[Nested comments]	Kommentare können ineinander verschachtelt werden.
		
		Beispiel:
		`(*`
		`  a := inst.out; (* to be checked *)`
		`  b := b+1;`
		`*)`
		Der Kommentar, der mit der ersten Klammer beginnt, wird hier nicht bereits durch die erste schließende Klammer nach `checked` abgeschlossen, sondern erst durch die zweite schließende Klammer.

[Binärfile erzeugen]	[Create binary file of the application]	Beim Übersetzen wird ein binäres Abbild des erzeugten Codes (Boot-Projekt) im Projektverzeichnis angelegt (Dateiname: `<projektname>.bin`). Mit dem Menübefehl [Online] / [Bootprojekt erzeugen] ([Online] / [Create boot project]) haben Sie die Möglichkeit, das Bootprojekt und die Binärdatei mit der zugehörigen Checksumme online auf der Steuerung bzw. offline im Projektverzeichnis abzulegen.

[Aktionen verschatten Programme]	[Actions hide programs]	Die Option [Aktionen verschatten Programme] regelt die Abarbeitungsreihenfolge , wenn eine lokale Aktion den gleichen Namen hat wie eine Variable oder ein Programm: 1. lokale Variable 2. lokale Aktion 3. globale Variable 4. Programm Die Option ist standardmäßig aktiviert. Wenn Sie die Option deaktivieren, ist die Abarbeitungsreihenfolge: 1. lokale Variable 2. globale Variable 3. Programm 4. lokale Aktion **Achtung**: Wenn ein Projekt geöffnet wird, das mit einer früheren Version des PLC-Editors erstellt wurde, ist die Option standardmäßig deaktiviert.
[LREAL als REAL übersetzen]	[Treat LREAL as REAL]	Die Option [LREAL als REAL übersetzen] regelt, ob LREAL-Werte beim Übersetzen des Projekts wie REAL-Werte behandelt werden. Damit können Sie plattformunabhängige Projekte erstellen. Die Verfügbarkeit ist abhängig vom Luafzeitsystem. Die Option ist standardmäßig deaktiviert.
[Anzahl der Datensegmente]	[Number of data segments]	Mit der Angabe [Anzahl der Datensegmente] legen Sie fest, wie viele Speichersegmente in der Steuerung für die Daten des Projekts reserviert werden sollen. Dieser Platz ist nötig, damit ein **Online Change** durchgeführt werden kann, nachdem neue Variablen hinzugefügt wurden. Wenn Sie beim Übersetzen die Meldung "Speicher für globale Variablen aufgebraucht." erhalten, erhöhen Sie die hier eingetragene Anzahl. Lokale Programmvariablen werden in dieser Beziehung wie globale Variablen behandelt.

[Objekte ausschließen]	[Exclude objects]	Nach dem Klicken auf die Schaltfläche [Objekte ausschließen], wird der Dialog [Objekte vom Übersetzen ausschließen] ([Exclude objects from build]) geöffnet: In diesem Dialog legen Sie fest, welche Objekte nicht übersetzt werden sollen: 1. Klicken Sie auf die gewünschten Bausteine. Sie können die <Umschalt>- und die <Strg>-Taste benutzen, um mehrere Bausteine gleichzeitig auszuwählen. 2. Aktivieren Sie die Option [Nicht übersetzen] ([Exclude]). Die Bausteine werden daraufhin in grüner Farbe dargestellt. Um automatisch alle Bausteine auszuschließen, die im Programm nicht verwendet werden, klicken Sie auf die Schaltfläche [Unbenutzte ausschließen] ([Exclude unused]). **Hinweis**: Sie können auch im **Object Organizer** ein markiertes Objekt vom Übersetzen ausschließen, indem Sie den Befehl [Vom Übersetzen ausschließen] ([Exclude from build]) im Kontextmenü ausführen.
[Makro vor/nach dem Übersetzen]	[Macro before/after compile]	Um auf den Übersetzungsvorgang Einfluss zu nehmen, können Sie zwei Makros angeben: Das Makro im Feld [Makro vor dem Übersetzen] wird vor dem Übersetzungslauf ausgeführt, das Makro im Feld [Makro nach dem Übersetzen] danach. Folgende Makrobefehle können hier nicht ausgeführt werden: • file new, file open, file close, file saveas, file quit • online • project compile, project check, project build, project clean, project rebuild • debug, watchlist
[Compiler-Version]	[Compiler version]	Die für den Übersetzungsvorgang zu verwendende Compiler-Version kann hier definiert werden. Es stehen jeweils die aktuelle und die bisherigen Compiler-Versionen (für jede Version / jedes Service Pack / jedes Patch) zurückgehend bis V2.3.3 zur Verfügung. Wenn Sie ein Projekt stets mit der neuesten Compiler-Version übersetzen wollen, aktivieren Sie die Option [Immer Aktuell] ([Use latest]). Wenn das Projekt automatisch mit einer bestimmten Compiler-Version übersetzt werden soll, stellen Sie diese über das Auswahlfeld [Festlegen] ([Fix]) ein.
[Automatisch prüfen]	[Check automatically]	Folgende Optionen können Sie für die Überprüfung der semantischen Korrektheit aktivieren: • [Unbenutzte Variablen] ([Unused variables]) • [Überlappende Speicherbereiche] ([Overlapping memory areas]) • [Konkurrierender Zugriff] ([Concurrent access]) • [Mehfaches Speichern auf Output] ([Multiple write access on output])

4.2.10 [Kennworte] ([Passwords])



20265AEN

Um eine Projektdatei vor unerwünschten Zugriffen zu schützen, können Sie sowohl das Öffnen als auch das Verändern der Datei durch Kennworte absichern.

Geben Sie das gewünschte Kennwort im Eingabefeld [Kennwort] ([Password]) ein. Für jedes eingegebene Zeichen erscheint ein Sternchen (*). Wiederholen Sie die Kennwort-Eingabe im Eingabefeld [Kennwort bestätigen] ([Confirm Password]). Schließen Sie den Dialog mit [OK].

Wenn die Meldung "Das Kennwort und seine Bestätigung stimmen nicht überein." erscheint, wiederholen Sie beide Einträge, bis der Dialog korrekt geschlossen wird.

Speichern Sie die Projektdatei.

Beim nächsten Öffnen der Projektdatei erscheint ein Dialog zur Eingabe des Kennworts. Das Projekt wird nur geöffnet, wenn Sie das richtige Kennwort eingeben. Bei falscher Eingabe des Kennworts erscheint die Meldung "Das Kennwort ist nicht korrekt."

Sie können die Projektdatei auch zusätzlich vor Veränderungen mit einem Kennwort schützen. Geben Sie hierzu ein Kennwort in das Feld [Schreibschutz-Kennwort] ([Write Protection Password]) ein, und wiederholen Sie das Kennwort im Feld [Schreibschutz-Kennwort bestätigen] ([Confirm Write Protection Password]).

Ein schreibgeschütztes Projekt kann auch ohne Kennwort geöffnet werden. Klicken Sie hierzu bei Abfrage des Schreibschutz-Kennworts auf die Schaltfläche [Abbrechen] ([Cancel]). Sie können ein schreibgeschützt geöffnetes Projekt übersetzen, in die Steuerung laden, simulieren usw. Sie können es jedoch nicht verändern.

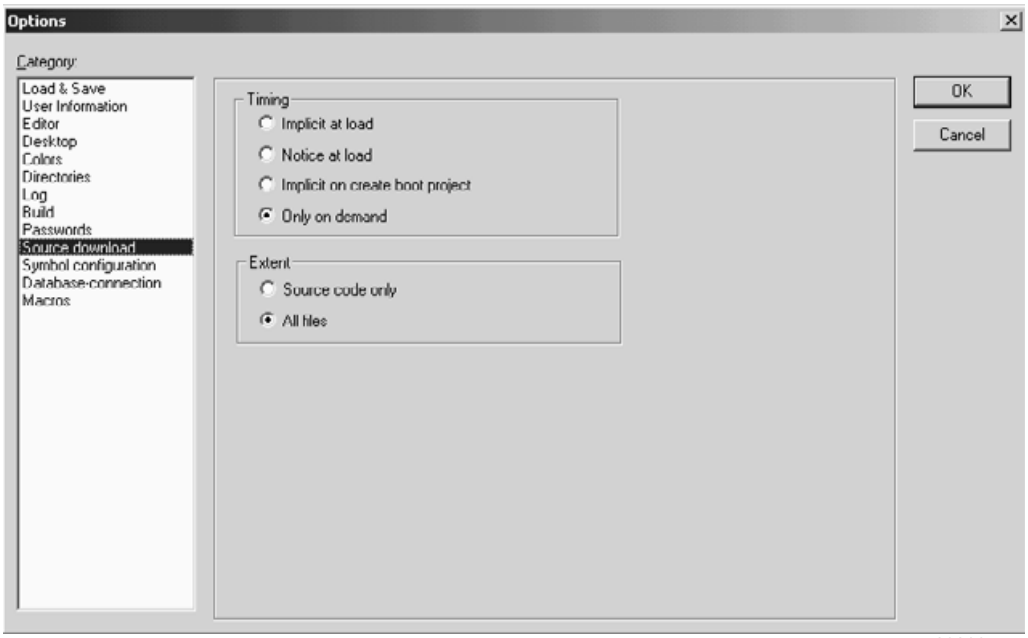
Wenn Sie ein Kennwort vergessen haben, wenden Sie sich bitte an Ihren Steuerungshersteller.

Die Kennworte werden mit dem Projekt gespeichert.

Um differenziertere Zugriffsrechte zu schaffen, können Sie Arbeitsgruppen über folgende Menübefehle festlegen:

- [Projekt] / [Objekt] / [Eigenschaften] ([Project] / [Object] / [Properties]), Seite 161
- [Projekt] / [Passwörter für Arbeitsgruppen] ([Project] / [User group passwords]), Seite 145

4.2.11 [Sourcedownload] ([Source download])

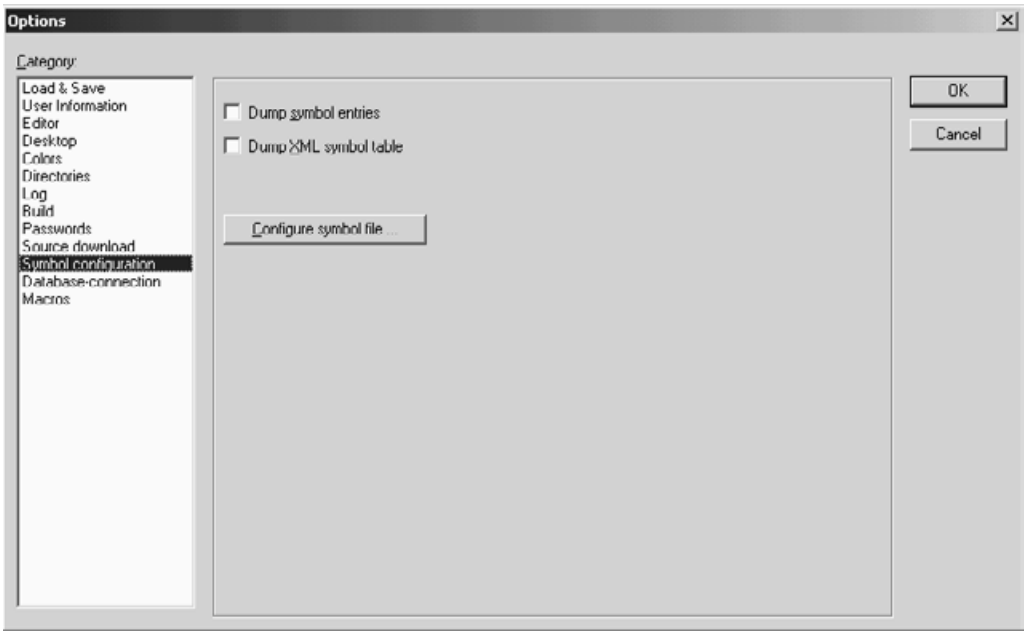


20266AEN

Sie können einstellen, wann und in welchem Umfang der Quellcode des Projekts in die Steuerung gespeichert wird. Die Daten werden hierzu gepackt.

[Zeitpunkt]	**[Timing]**	Die in diesem Bereich gemachten Einstellungen beziehen sich immer auf die im Bereich [Umfang] ausgewählten Dateien.
[Implizit beim Laden]	[Implicit at load]	Der Quellcode wird nach Ausführen des Menübefehls [Online] / [Laden] ([Online] / [Download]) in die Steuerung geladen.
[Hinweis beim Laden]	[Notice at load]	Nach Ausführen des Menübefehls [Online] / [Laden] erscheint der Dialog [Quellcode in die Steuerung schreiben?]. Wenn Sie auf [Ja] ([Yes]) klicken, wird der Quellcode in die Steuerung geladen.
[Implizit beim Bootprojekt erzeugen]	[Implicit on create boot project]	Der Quellcode wird nach Ausführen des Menübefehls [Online] / [Bootprojekt erzeugen] ([Online] / [Create boot project]) in die Steuerung geladen.
[Nur auf Anforderung]	[Only on demand[	Das Laden des Quellcodes in die Steuerung ist nur über den Menübefehl [Online] / [Quellcode laden] ([Online] / [Sourcecode download]) möglich.
[Umfang]	**[Extent]**	
[Nur Quellcode]	[Source code only]	Die im Bereich [Zeitpunkt] gemachten Einstellungen gelten nur für die Projektdatei (Datei-Erweiterung `.pro`).
[Alle Dateien]	[All files]	Neben der Projekt-Datei werden weitere Dateien abgespeichert, wie z. B. zugehörige Bibliotheken, Visualisierungs-Bitmaps, Konfigurationsdateien.

4.2.12 [Symbolkonfiguration] ([Symbol configuration])



20267AEN

Die Kategorie [Symbolkonfiguration] dient der Konfiguration der Symboldatei, die bei jedem Übersetzen des Projekts erzeugt wird. Die Kategorie ist im Simulationsmodus nicht verfügbar.

Die Symboldatei wird in Abhängigkeit vom verwendeten Gateway-Server als Textdatei `<Projektname>.sym` oder als Binärdatei `<Projektname>.sdb` im Projektverzeichnis angelegt. Die Symboldatei ist für den Datenaustausch mit der Steuerung über die Symbolschnittstelle nötig und wird beispielsweise vom OPC- oder GatewayDDE-Server verwendet.

[Symboleinträge erzeugen]	[Dump symbol entries[	Wenn die Option [Symboleinträge erzeugen] aktiviert ist, werden bei jedem Übersetzen des Projekts automatisch Symboleinträge für die Projektvariablen in der Symboldatei angelegt. Wenn diese Option deaktiviert ist, enthält die Symboldatei nur Versionsinformationen zur Symboldatei selbst und zum Projekt sowie eine Checksumme.
[XML-Symboltabelle erzeugen]	[Dump XML symbol table]	Es wird eine XML-Version (`<Projektname>.SYM_XML`) der Symboldatei erzeugt.
[Symbolfile konfigurieren]	[Configure symbol file]	Unter [Symbolfile konfigurieren] können Sie die Symboldatei-Einträge konfigurieren.

Die Konfiguration der Symboldatei-Einträge kann auf zwei Arten erfolgen:

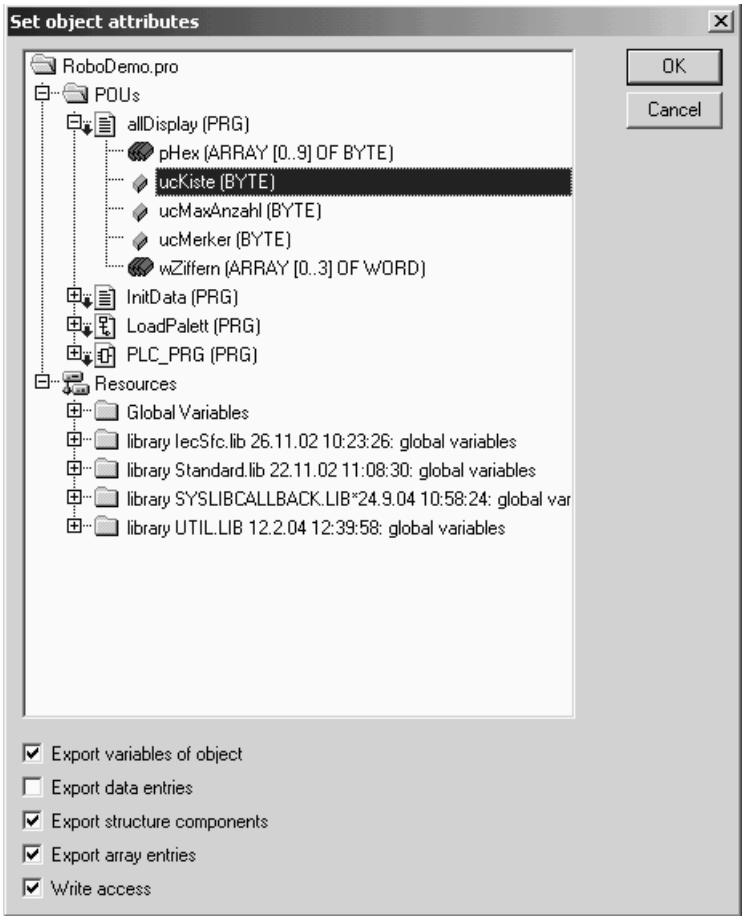
- Die Konfiguration wird aus der Datei `PLCEditor.ini` oder einer dort angegebenen, anderen `*.ini`-Datei eingelesen.
- Die Konfiguration erfolgt mit den unter [Symbolfile konfigurieren] angegebenen Einstellungen.

Wenn die Konfiguration der Symboldatei-Einträge aus einer `*.ini`-Datei eingelesen werden soll, gehen Sie wie folgt vor:

1. Klicken Sie im **Object Organizer** auf die Registerkarte [Ressourcen] ([Resources]).
2. Doppelklicken Sie auf [Zielsystemeinstellungen] ([Target settings]).
3. Aktivieren Sie in der Registerkarte [Allgemein] ([General]) die Option [Symbolkonfiguration aus INI-Datei] ([Symbol config from INI file]).
4. Bestätigen Sie mit [OK].

Wenn die Konfiguration nicht über eine `*.ini`-Datei erfolgen soll, klicken Sie auf die Schaltfläche [Symbolfile konfigurieren]. Der Dialog [Objektattribute setzen] ([Set object

attributes]) wird geöffnet.



20277AEN

Wählen Sie in der Baumstruktur die Variablen aus, für die Symboleinträge erzeugt werden sollen. Durch Klicken auf einen Baustein oder eine Ressource markieren Sie automatisch alle zugehörigen Variablen. Sie können auch einzelne Variablen auswählen. Mit der <Umschalt>-Taste und der <Strg>-Taste markieren Sie mehrere Bausteine/Variablen gleichzeitig. Für die markierten Objekte stehen folgende Optionen zur Verfügung:

[Variablen des Objekts ausgeben]	[Export variables of object]	Aktivieren Sie die Option [Variablen des Objekts ausgeben], damit Sie die anderen Optionen einstellen können.
[Sammeleinträge ausgeben]	[Export data entries]	Für Strukturen und Arrays des Objekts werden Einträge zum Zugriff auf die Gesamtvariablen erzeugt.
[Strukturkomponenten ausgeben]	[Export structure components]	Für jede Variablenkomponente einer Struktur des Objekts wird ein eigener Eintrag erzeugt.
[Feldkomponenten ausgeben]	[Export array entries]	Für jede Variablenkomponente eines Arrays des Objekts wird ein eigener Eintrag erzeugt.
[Schreibzugriff]	[Write access]	Die Variablen dürfen vom OPC-Server geändert werden.

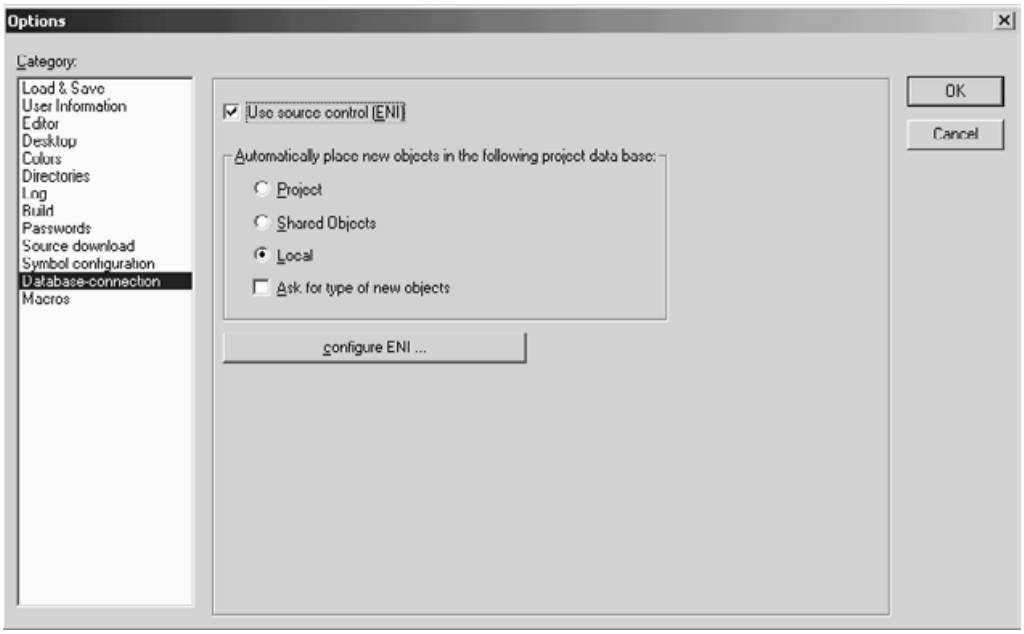
Nachdem Sie die Einstellungen für die aktuelle Variablenauswahl vorgenommen haben, können Sie – ohne den Dialog vorher mit [OK] schließen zu müssen – andere Bausteine auswählen und konfigurieren.

Nachdem Sie alle Einstellungen vorgenommen haben, beenden Sie den Dialog mit [OK]. Die Einstellungen werden übernommen.

Hinweis:

Mit Hilfe von Pragma-Anweisungen können Sie für einzelne Variablen festlegen, ob die Variablen ohne Schreib- bzw. Leserecht oder gar nicht in die Symboldatei übernommen werden.

4.2.13 [Projektdatenbank] ([Database-connection])



20268AEN

In dier Kategorie [Projektdatenbank] legen Sie fest, ob und wie das Projekt in einer Projektdatenbank verwaltet werden soll. Dazu konfigurieren Sie die ENI-Schnittstelle.

[Projektdatenbank (ENI) verwenden]	[Use source control (ENI)]	Aktivieren Sie die Option [Projektdatenbank (ENI) verwenden], wenn Sie über einen ENI-Server auf eine Projektdatenbank zugreifen wollen, um alle oder bestimmte zum Projekt gehörigen Bausteine über diese Projektdatenbank zu handhaben. Voraussetzung hierfür ist, dass ENI-Server und Projektdatenbank installiert sind und Sie als gültiger Datenbank-Benutzer definiert sind. Weitere Informationen hierzu finden Sie in der Dokumentation zum ENI-Server. Wenn diese Option aktiviert ist, stehen für jedes Objekt des Projekts die Funktionen der Projektdatenbank zur Verfügung. Einige Datenbankfunktionen können so konfiguriert werden, dass sie automatisch ablaufen. Die Datenbankfunktionen können Sie auch manuell über den Menübefehl [Projekt] / [Projektdatenbank] ([Project] / [Project database]) aufrufen.
[Neue Objekte automatisch in folgende Projektdatenbank ablegen]	**[Automatically place new objects in the following project data base]**	Wenn ein neues Objekt im Projekt eingefügt wird, erfolgt eine automatische Zuordnung zu der hier eingestellten Projektdatenbank. Ein Objekt kann jeweils nur einer Projektdatenbank zugeordnet werden. Diese Zuordnung können Sie über den Menübefehl [Projekt] / [Objekt] / [Eigenschaften] ([Project] / [Object] / [Properties]) abfragen und ändern. **Hinweis**: Es gibt noch die Projektdatenbank "Übersetzungsdateien" für Objekte, die erst beim Kompilieren eines Programms entstehen. Die Einstellungen für diese Projektdatenbank werden unter [ENI konfigurieren] vorgenommen.
[Projekt]	[Project]	Das Objekt wird im ENI-Datenbankverzeichnis abgelegt. Das Datenbankverzeichnis legen Sie unter [ENI konfigurieren] fest.
[Gemeinsame Objekte]	[Shared objects]	Das Objekt wird im ENI-Datenbankverzeichnis verwaltet. Das Datenbankverzeichnis legen Sie unter [ENI konfigurieren] fest.
[Lokal]	[Local]	Das Objekt wird nicht über die ENI-Schnittstelle in der Projektdatenbank verwaltet, sondern ausschließlich lokal im Projekt gespeichert. **Hinweis**: Objekte, die einer Projektdatenbank zugeordnet sind, werden in jedem Fall auch lokal abgespeichert.
[Nach Art von neuen Objekten fragen]	[Ask for type of new objects]	Die aktivierte Option [Nach Art von neuen Objekten fragen] schaltet die automatische Zuordnung eines Objekts zu einer Projektdatenbank ab. Bei jedem Einfügen eines neuen Objekts wird der Dialog [Eigenschaften] ([Properties]) geöffnet, in dem Sie die Zuordnung zu einer Projektdatenbank festlegen.

[ENI konfigurieren]	[configure ENI]	Die Schaltfläche öffnet einen Dialog, in dem Sie die Einstellungen für die drei Projektdatenbanken [Projektobjekte], [Gemeinsame Objekte] und [Übersetzungsdateien] vornehmen.

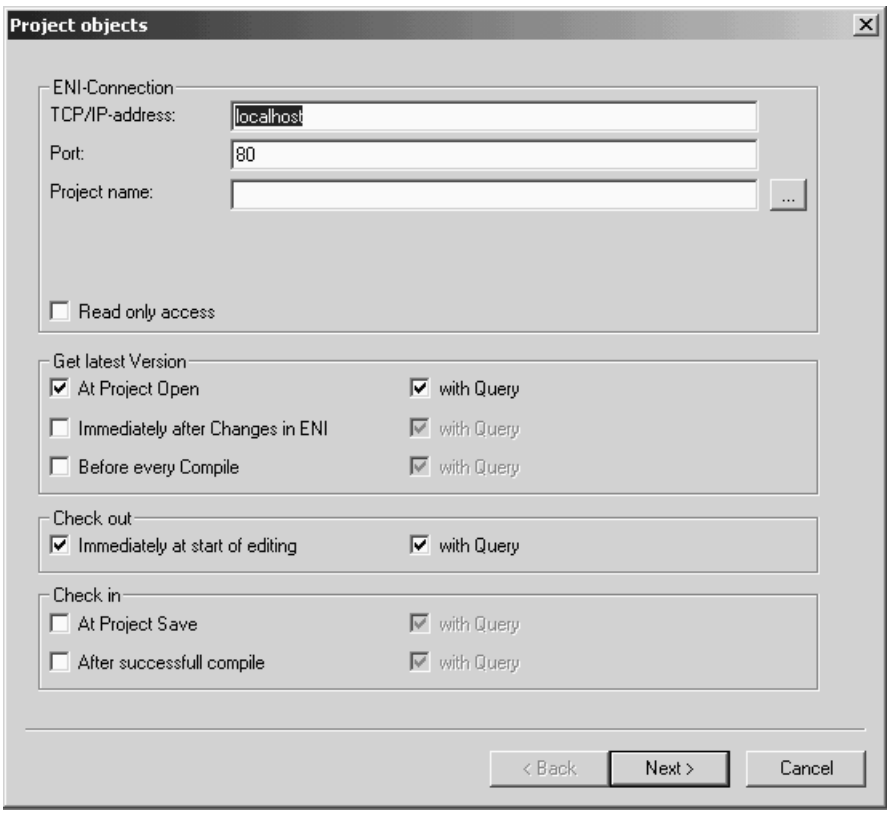
ENI konfigurieren

Der Aufbau des Dialogs hängt davon ab, ob die ENI-Konfiguration schon einmal ausgeführt wurde:

- Beim ersten Aufrufen der ENI-Konfiguration werden die Einstellungen für die drei Projektdatenbanken nacheinander abgefragt. Nach dem Klicken auf die Schaltfläche [Weiter] ([Next]) erfolgt der Wechsel zur nächsten Projektdatenbank. Beim Wechsel vom Dialog [Projektobjekte] ([Project objects]) zum Dialog [Gemeinsame Objekte] ([Shared objects]) werden die Einträge übernommen.
- Bei jedem weiteren Aufruf der ENI-Konfiguration erscheint ein Dialog mit drei Registerkarten.

Wenn Sie sich vor der Konfiguration nicht bereits über den Menübefehl [Projekt] / [Projektdatenbank] / [Login] ([Project] / [Project database] / Login]) in die Datenbank eingeloggt hatten, erscheint vorher der Dialog [Login] ([Login]).

Dialoge [Projektobjekte] und [Gemeinsame Objekte] ([Project objects] und [Shared objects])

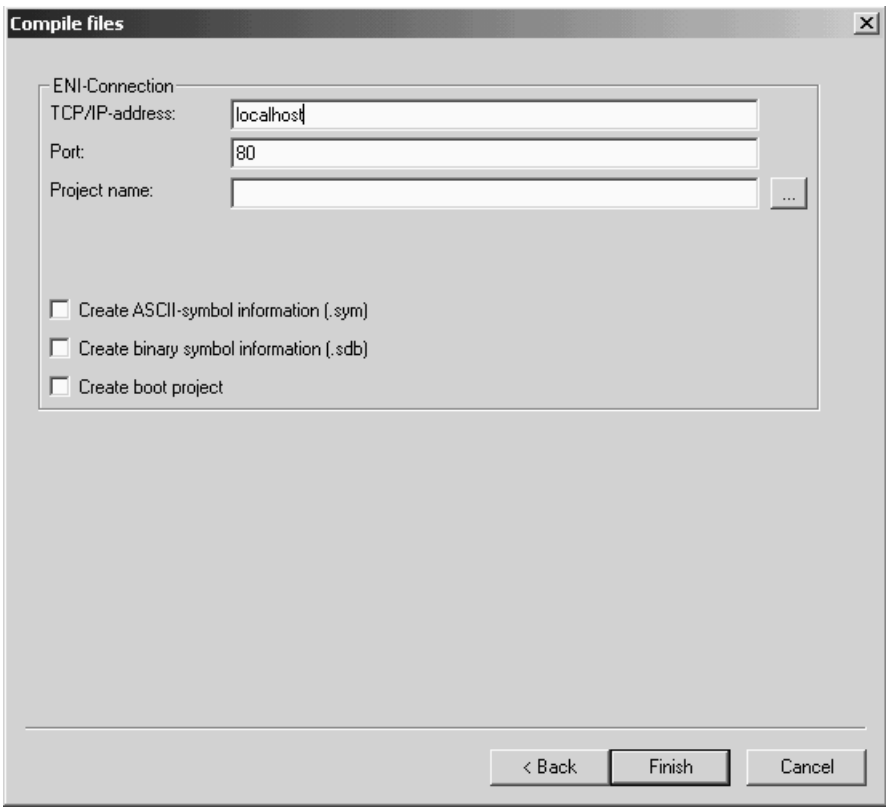


20278AEN

In den beiden Dialogen definieren Sie, mit welchen Zugangsparametern die Objekte der Projektdatenbanken [Projektobjekte] und [Gemeinsame Objekte] in der Datenbank verwaltet werden. Die Dialoge sind für beide Projektdatenbanken gleich.

[ENI-Verbindung]	**[ENI-Connection]**	
[TCP/IP-Adresse]	[TCP/IP-address]	Adresse des Rechners, auf dem der ENI-Server läuft.
[Port]	[Port]	Default: 80; Der Wert muss mit der Einstellung in der ENI-Server-Konfiguration übereinstimmen.
[Projektname]	[Project name]	Der Name des Verzeichnisses in der Datenbank, in dem die Objekte der betreffenden Kategorie abgelegt werden sollen. Wenn das Verzeichnis in der Datenbank bereits existiert, können Sie es im Verzeichnisbaum der ENI-Projekte auswählen, den Sie über die Schaltfläche [...] öffnen. **Hinweis**: Wenn Sie sich vorher noch nicht über den Login-Dialog als ENI-Benutzer eingeloggt haben, erscheint beim Drücken dieser Schaltfläche zunächst der Dialog [Login] ([Login]), in dem Sie Benutzername und Passwort für den ENI-Zugang zu den drei Projektdatenbanken eingeben müssen.
[Nur lesender Zugriff]	[Read only access]	Auf die Daten des unter [Projektname] definierten Datenbankverzeichnisses kann nur lesend zugegriffen werden.
[Abrufen]	**[Get latest Version]**	Wenn sich die Projektdatenbank-Version eines Objekts von der im lokal geöffneten Projekt unterscheidet, kopiert die Datenbankfunktion [Abrufen] die Datenbankversionen ins lokal geöffnete Projekt. Dadurch wird die lokale Projektversion überschrieben. In diesem Bereich legen Sie fest, zu welchem Zeitpunkt das Abrufen erfolgen soll.
[Beim Projekt öffnen]	[At Project Open]	Abrufen, wenn das Projekt im PLC-Editor geöffnet wird.
[Sofort bei Änderungen im ENI]	[Immediately after Changes in ENI]	Abrufen, wenn in der Datenbank eine neuere Version eines Bausteins eingecheckt wird. Der Baustein wird unmittelbar im geöffneten Projekt aktualisiert und eine entsprechende Meldung ausgegeben.
[Vor jedem Compile]	[Before every Compile]	Vor jedem Kompilieren (Übersetzen) des Projekts abrufen.
[mit Nachfrage]	[with Query]	Das Abrufen wird erst auf Nachfrage ausgeführt. Im erscheinenden Dialog können Sie das Abrufen bestätigen bzw. abbrechen.
[Auschecken]	**[Check out]**	Die Datenbankfunktion [Auschecken] bedeutet, dass das Objekt als "in Bearbeitung" markiert wird und für andere Benutzer gesperrt ist. Das Objekt wird wieder freigegeben, wenn Sie das Objekt einchecken oder das Auschecken rückgängig machen..
[Unmittelbar bei Beginn einer Änderung]	[Immediately at start of editing]	Das Auschecken eines Objekts erfolgt automatisch, sobald mit dessen Bearbeitung im Projekt begonnen wird. Wenn das Objekt bereits durch einen anderen Benutzer ausgecheckt ist (erkenntlich an einem roten Kreuz vor dem Objektnamen im **Object Organizer**), wird eine Meldung ausgegeben.
[mit Nachfrage]	[with Query]	Das Auschecken wird erst auf Nachfrage ausgeführt. Im erscheinenden Dialog können Sie das Auschecken bestätigen bzw. abbrechen.
[Einchecken]	**[Check in]**	Die Datenbankfunktion [Einchecken] bedeutet, dass eine neue Version eines Objekts in der Datenbank angelegt wird. Die alten Versionen bleiben erhalten. In diesem Bereich legen Sie fest, zu welchem Zeitpunkt das Einchecken erfolgen soll.
[Bei Projekt Speichern]	[At Project Save]	Jedes veränderte Objekt wird automatisch bei jedem Speichern des Projekts eingecheckt.
[Nach erfolgreichem Compile]	[After successful compile]	Nach jedem fehlerfreien Übersetzen des Projekts wird jedes veränderte Objekt eingecheckt.
[mit Nachfrage]	[with Query]	Das Einchecken wird erst auf Nachfrage ausgeführt. Im erscheinenden Dialog können Sie das Einchecken bestätigen bzw. abbrechen.

Dialog [Übersetzungsdateien] ([Compile files])

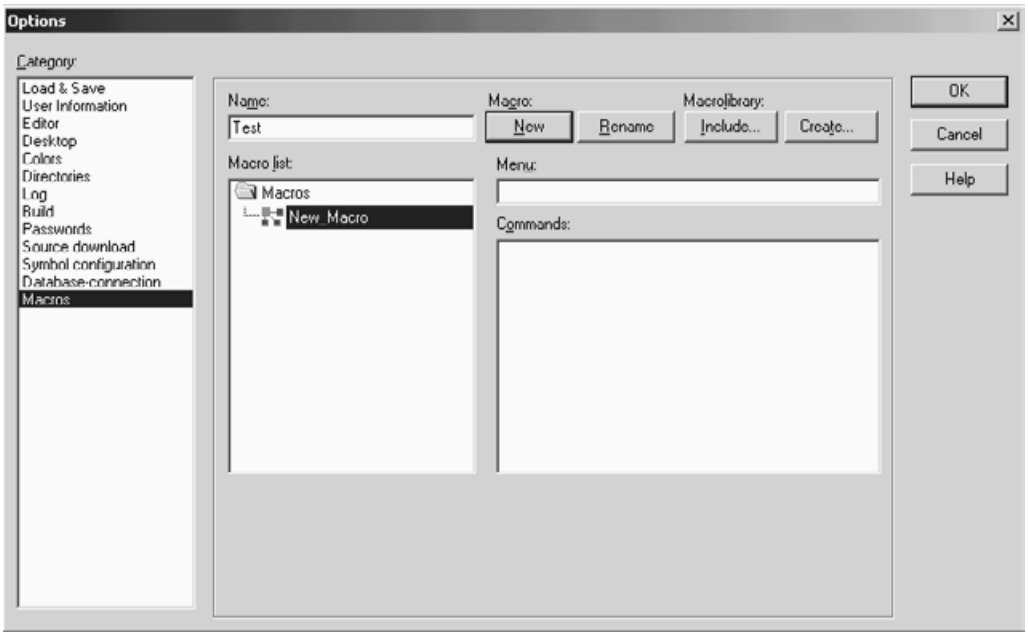


20203AEN

In dem Dialog [Übersetzungsdateien] definieren Sie, wie Übersetzungsdateien in der Datenbank verwaltet werden.

[ENI-Verbindung]	**[ENI-Connection]**	
[TCP/IP-Adresse]	[TCP/IP-address]	Adresse des Rechners, auf dem der ENI-Server läuft.
[Port]	[Port]	Default: 80; Der Wert muss mit der Einstellung in der ENI-Server-Konfiguration übereinstimmen.
[Projektname]	[Project name]	Name des Verzeichnisses in der Datenbank, in dem die Objekte der betreffenden Kategorie abgelegt werden sollen. Falls das Verzeichnis in der Datenbank bereits existiert, können Sie es im Verzeichnisbaum der ENI-Projekte auswählen, den Sie über die Schaltfläche [...] öffnen. **Hinweis**: Wenn Sie sich vorher nicht in die ENI-Projektdatenbank eingeloggt hatten, erscheint nach dem Klicken auf die Schaltfläche zunächst der Dialog [Login] ([Login]), in dem Sie Benutzername und Passwort für den ENI-Zugang zu den drei Projektdatenbanken eingeben müssen.
[ASCII-Symbolinformation erzeugen (.sym)]	[Create ACII-symbol information (.sym)]	Die Symboldatei wird im Textformat (Dateiendung `.sym`) in die Datenbank geschrieben. Beim Erzeugen der Symbole gelten die in den Projektoptionen in der Kategorie [Symbolkonfiguration] ([Symbol configuration]) gesetzten Objektattribute.
[Binär-Symbolinformation erzeugen (.sdb)]	[Create binary symbol information (.sdb)]	Die Symboldatei wird im Binärformat (Dateiendung `.sdb`) in die Datenbank geschrieben. Beim Erzeugen der Symbole gelten die in den Projektoptionen in der Kategorie [Symbolkonfiguration] gesetzten Objektattribute.
[Bootprojekt erzeugen]	[Create boot project]	Das Bootprojekt wird in der Projektdatenbank abgelegt.

4.2.14 [Makros] ([Macros])



20269AEN

In der Kategorie [Makros] können Sie mit Hilfe der Kommandodatei-Befehle des Batch-Mechanismus Makros definieren. Diese Makros sind anschließend über den Menübefehl [Bearbeiten] / [Makros] ([Edit] / [Macros]) abrufbar.

Die Makroeinträge erscheinen in der Reihenfolge ihrer Definition unter dem Menübefehl [Bearbeiten] / [Makros]. Eine Prüfung des Makros erfolgt erst beim Ausführen des Menübefehls.

Ein neues Makro definieren

1. Tragen Sie im Eingabefeld [Name] ([Name]) einen Namen für das zu erstellende Makro ein. Nach dem Klicken auf die Schaltfläche [Neu] ([New]) wird dieser Name in die Makroliste übernommen und dort markiert. Die Makroliste ist in einer Baumstruktur angelegt. Die lokal angelegten Makros stehen untereinander, eventuell eingebundene Makrobibliotheken (siehe unten) erscheinen mit dem Namen der Bibliotheksdatei. Über das Plus- bzw. Minuszeichen vor dem Bibliotheksnamen kann die Liste der Bibliothekselemente auf- bzw. zugeklappt werden.
2. Definieren Sie im Feld [Menü] ([Menu]), wie der Menüeintrag heißen soll, mit dem das Makro im Menü [Bearbeiten] / [Makros] eingetragen wird. Um einen Buchstaben als Shortcut zu definieren, müssen Sie diesem Buchstaben das Zeichen "&" vorangestellen. Beispiel: Der Name "Ma&kro 1" erzeugt den Menüeintrag "Makro 1".
3. Geben Sie im Editorfeld [Anweisungen] ([Commands]) die Kommandos für das in der Makroliste markierte Makro ein. Alle Kommandos und Schlüsselwörter des Batch-Mechanismus des PLC-Editors sind zulässig.

   Über die Schaltfläche [Hilfe] ([Help]) erhalten Sie eine Auflistung der Makrobefehle.

   Eine neue Anweisungszeile fügen Sie mit dem Tastaturbefehl <Strg>+<Eingabetaste> eingefügt. Über die rechte Maustaste erhalten Sie das Kontextmenü mit den üblichen Texteditorfunktionen. Zusammengehörige Kommando-Bestandteile können mit Anführungszeichen zusammengefasst werden.
4. Falls Sie weitere Makros anlegen wollen, wiederholen Sie die Schritte 1 bis 3.
5. Bestätigen und beenden Sie den Dialog mit [OK]. Die aktuelle Beschreibung der Makros wird im Projekt gespeichert.

Ein Makro löschen

1. Markieren Sie das Makro in der Makroliste.
2. Drücken Sie die <Entf>-Taste.

Ein Makro umbenennen

1. Markieren Sie das Makro in der Makroliste.
2. Geben Sie unter [Name] ([Name]) einen anderen Namen ein.
3. Klicken Sie auf die Schaltfläche [Umbenennen] ([Rename]).

Ein vorhandenes Makro bearbeiten

1. Markieren Sie das Makro in der Makroliste.
2. Editieren Sie die Eingabefelder [Menü] ([Menu]) und/oder [Anweisungen] ([Commands]).
3. Wiederholen Sie bei Bedarf die Schritte 1 und 2 für weitere Makros.
4. Beenden und bestätigen Sie die Änderungen mit [OK]. Die aktuelle Beschreibung der Makros wird im Projekt gespeichert.

Makro-bibliotheken

Makros können in externen Makrobibliotheken gespeichert werden, um sie in andere Projekte einzubinden.

Eine Makrobibliothek aus den Makros des aktuellen Projekts erstellen

1. Klicken Sie auf die Schaltfläche [Erstellen] ([Create]). Der Dialog [Objekte kopieren] ([Merge Project]) wird geöffnet. In dem Dialog sind alle verfügbaren Makros aufgelistet.
2. Markieren Sie die gewünschten Makros. Mit den Tasten <Strg> und <Umschalt> können Sie mehrere Makros gleichzeitig markieren.
3. Bestätigen Sie mit [OK]. Daraufhin wird der Dialog [Objekte kopieren] geschlossen und der Dialog [Makrobibliothek speichern] ([Save macro library]) geöffnet.
4. Geben Sie hier Namen und Verzeichnispfad für die zu erstellende Bibliothek ein und klicken Sie auf die Schaltfläche [Speichern] ([Save]). Die neue Bibliothek wird unter dem Namen `<bibliotheksname>.mac` angelegt und der Dialog [Makrobibliothek speichern] geschlossen.

Eine Makrobibliothek in das aktuelle Projekt einbinden

1. Klicken Sie auf die Schaltfläche [Einbinden] ([Include]). Der Dialog [Makrobibliothek öffnen] ([Open macro library]) wird geöffnet. Der Dialog zeigt automatisch nur Dateien mit der Erweiterung `.mac` an.
2. Markieren Sie die gewünschte Makrobibliothek.
3. Klicken Sie auf die Schaltfläche [Öffnen] ([Open]). Der Dialog [Makrobibliothek öffnen] wird geschlossen und die Makrobibliothek erscheint in der Makroliste.

Hinweis:

Ein Projekt kann mit Makros exportiert werden: Menübefehl [Projekt] / [Exportieren] ([Project] / [Export]).

4.3 Projekte verwalten

Die Menübefehle, die sich auf ein ganzes Projekt beziehen, stehen unter den Menüpunkten [Datei] und [Projekt] ([File] und [Project]). In diesem Kapitel werden die Menübefehle zum Verwalten von Projekten näher beschrieben.

4.3.1 [Datei] / [Neu] ([File] / [New])

Schaltfläche: [ ]

Mit dem Menübefehl [Datei] / [Neu] legen Sie ein leeres Projekt mit dem Namen "Unbenannt" an. Diesen Namen müssen Sie beim Speichern ändern.

4.3.2 [Datei] / [Neu aus Vorlage] ([File] / [New from template])

Mit dem Menübefehl [Datei] / [Neu aus Vorlage] können Sie ein beliebiges Projekt als Vorlage verwenden. Das Projekt muss vorher nicht mit speziellen Einstellungen für diesen Zweck abgespeichert worden sein.

Nach dem Ausführen des Menübefehls wird der Dialog [Öffnen] ([Open]) zur Auswahl einer Projektdatei als Vorlage geöffnet. Das neue Projekt hat dann den Namen "Unbenannt". Diesen Namen müssen Sie beim Speichern ändern.

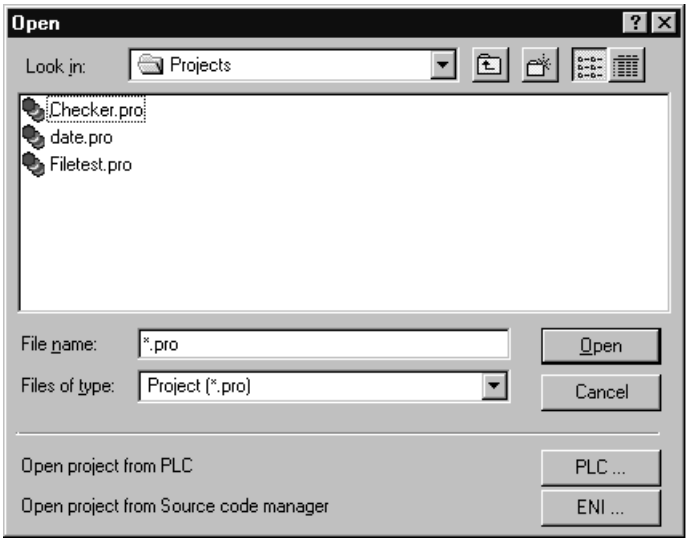
4.3.3 [Datei] / [Öffnen] ([File] / [Open])

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<O>

Mit dem Menübefehl [Datei] / [Öffnen] öffnen Sie ein bestehendes Projekt. Wenn ein anderes Projekt im PLC-Editor geöffnet ist, wird dieses vorher geschlossen. Wenn das aktuell geöffnete Projekt seit dem letzten Speichern geändert wurde, wird ein Dialog mit der Meldung "Das Projekt wurde verändert. Vor dem Schließen speichern?" ("The project has changed. Save it before closing?") geöffnet. Wenn Sie auf [Ja] ([Yes]) klicken, werden die Änderungen gespeichert.

Nach dem Ausführen des Menübefehls wird der Dialog [Öffnen] ([Open]) geöffnet. Wählen Sie eine Projektdatei (Erweiterung `.pro`) oder eine Bibliotheksdatei (Erweiterung `.lib`).



20282AEN

Hinweis:

Sie können mit dem Menübefehl [Öffnen] ([Open]) kein Projekt erzeugen.

Wenn für ein Projekt Kennwörter oder Arbeitsgruppen definiert sind, wird der Dialog zur Eingabe der Kennwörter geöffnet.

Projekt aus der Steuerung öffnen

Um eine Projektdatei aus einer Steuerung zu laden, klicken Sie auf die Schaltfläche [SPS] ([PLC]). Wenn noch keine Verbindung zur Steuerung besteht, wird zunächst der Dialog [Zielsystemeinstellungen] ([Target settings]) geöffnet. Wählen Sie unter [Konfiguration] ([Configuration]) das gewünschte Zielsystem aus und klicken Sie auf [OK].

Der Dialog [Kommunikationsparameter] ([Communication Parameters]) zum Einstellen der Übertragungsparameter wird geöffnet. Weitere Informationen dazu finden Sie auf Seite 190 (→ Kap. 4.6.22, [Online] / [Kommunikationsparameter] ([Online] / [Communication Parameters])).

Nach Herstellung der Online-Verbindung wird geprüft, ob gleichnamige Projektdateien bereits im Verzeichnis auf Ihrem Rechner vorliegen. Wenn dies der Fall ist, wird der Dialog [Projekt aus der Steuerung laden] ([Load project from controller]) geöffnet. In diesem Dialog können Sie entscheiden, ob die lokalen Dateien durch die in der Steuerung ersetzt werden sollen. Dieser Vorgang ist die Umkehrung des Menübefehls [Online] / [Quellcode laden] ([Online] / [Load source code]), mit dem Sie die Quelldatei des Projekts in der Steuerung speichern.

Hinweis:

Beachten Sie, dass nach dem Hochladen eines Projekts dieses noch keinen Namen hat. Sie müssen es unter einem neuem Namen abspeichern! Bei einigen Zielsystemen wird eine in der Projektinformation eingetragene Bezeichnung automatisch als neuer Dateiname vorgegeben. Bei diesen Zielsystemen wird beim Laden des Projekts aus der SPS automatisch der Dialog [Speichern] ([Save]) geöffnet.

Wenn sich auf der Steuerung kein Projekt befindet, erhalten Sie eine entsprechende Fehlermeldung.

Weitere Informationen finden Sie auf Seite 89 (→ Kap. 4.2.11, [Sourcedownload] ([Source download])).

Projekt aus Projektdatenbank öffnen

Die Option [Projekt aus Projektdatenbank öffnen] ([Open project from source code manager]) öffnet ein Projekt, das in einer ENI-Projektdatenbank verwaltet wird. Voraussetzung ist, dass Sie Zugang zu einem ENI-Server haben, der die Datenbank bedient.

1. Klicken Sie auf die Schaltfläche [ENI] ([ENI]). Der Dialog [Projektobjekte] ([Project objects]), der zum Aufbau der Verbindung zum ENI-Server dient, wird geöffnet.

   Geben Sie hier die Zugangsdaten ein:

   - [TCP/IP-Adresse] ([TCP/IP-address])
   - [Port] ([Port])
   - [Benutzername] ([User name])
   - [Passwort] ([Password])
   - [Projektname] ([Project name])
   - [Nur lesender Zugriff] ([Read only access])

2. Klicken Sie auf die Schaltfläche [Weiter] ([Next]). Der Dialog mit der Kategorie [Gemeinsame Objekte] ([Shared objects]) wird geöffnet.
3. Geben Sie auch hier die Zugangsdaten ein.
4. Klicken Sie auf die Schaltfläche [Fertigstellen] ([Finish]). Die Objekte aus den eingestellten Projekten werden abgerufen.

Sie können anschließend in den Projektoptionen die Einstellungen vornehmen, die für die weitere Bearbeitung des Projekts gelten sollen. Wenn Sie das Projekt weiterhin in der Projektdatenbank verwalten wollen, müssen Sie in den Projektoptionen die Kategorie [Projektdatenbank] ([Database-connection]) konfigurieren.

Die Zugangsdaten werden in der Datei `PLCEditor.ini` gespeichert. Benutzername und Passwort werden nur gespeichert, wenn in den Projektoptionen, Kategorie [Laden & Speichern] ([Load & Save]) die Option [Zugangsdaten für Projektdatenbank speichern] ([Save ENI credentials]) aktiviert ist.

Zuletzt geöffnete Projekte

Im Menü [Datei] ([File]) sind im Bereich unterhalb des Menüpunkts [Beenden] ([Exit]) die zuletzt bearbeiteten Projekte aufgelistet. Klicken Sie auf ein Projekt, um es zu öffnen.

4.3.4 [Datei] / [Schließen] ([File] / [Close])

Mit dem Menübefehl [Datei] / [Schließen] schließen Sie das aktuell geöffnete Projekt. Wenn das Projekt nach dem letzten Speichern verändert wurde, erscheint die Meldung: "Das Projekt wurde verändert. Vor dem Schließen speichern?"

Wenn das zu speichernde Projekt den Namen "Unbenannt" trägt, müssen Sie einen anderen Namen für das Projekt festlegen.

Weitere Informationen zum Speichern von Projekten finden Sie auf Seite 102 (→ Kap. 4.3.6, [Datei] / [Speichern unter] ([File] / [Save as])).

4.3.5 [Datei] / [Speichern] ([File] / [Save])

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<S>

Mit dem Menübefehl [Datei] / [Speichern] speichern Sie ein neues oder verändertes Projekt.

Wenn das zu speichernde Projekt den Namen "Unbenannt" trägt, müssen Sie einen anderen Namen für das Projekt festlegen.

Weitere Informationen zum Speichern von Projekten finden Sie auf Seite 102 (→ Kap. 4.3.6, [Datei] / [Speichern unter] ([File] / [Save as])).

4.3.6 [Datei] / [Speichern unter] ([File] / [Save as])

Mit dem Menübefehl [Datei] / [Speichern unter] speichern Sie das aktuelle Projekt unter einem anderen Projektnamen (`*.pro`) oder als Bibliothek (`*.lib`). Die ursprüngliche Projektdatei bleibt erhalten.

Nach dem Ausführen des Menübefehls wird der Dialog [Speichern unter] ([Save as]) geöffnet. Wählen Sie entweder ein existierendes Projekt aus, um es zu überschreiben, oder geben Sie einen neuen Dateinamen ein und wählen Sie den Dateityp.

Wenn Sie bei den Dateitypen eine frühere Version des PLC-Editor auswählen, gehen spezifische Daten der Version 2.3 verloren. Das Projekt kann dann aber mit der früheren Programmversion bearbeitet werden.

Sie können das aktuelle Projekt auch als Bibliothek abspeichern, um die erstellten Bausteine in anderen Projekten benutzen zu können. Wählen Sie dazu den Dateityp `Interne Bibliothek Version x.x (*.lib)`. "x.x" steht dabei alternativ für eine ältere Versionsnummer.

Wenn Sie Bausteine in anderen Programmiersprachen (z. B. C) implementiert haben und einbinden wollen, wählen Sie den Dateityp `Externe Bibliothek Version x.x (*.lib)`. Dadurch wird eine zusätzliche Datei abgespeichert, die den Dateinamen der Bibliothek mit der Erweiterung `.h` erhält. Diese Datei ist als C-Header-Datei aufgebaut und enthält die Deklarationen aller Bausteine, Datentypen und globalen Variablen. Bei externen Bibliotheken wird in der Simulation die Implementation ausgeführt, die im PLC-Editor zu den Bausteinen geschrieben wurde. Mit der echten Hardware wird die in der Programmiersprache C geschriebene Implementation abgearbeitet.

4.3.7 [Datei] / [Archiv speichern/versenden] ([File] / [Save/Mail Archive])

Mit dem Menübefehl [Datei] / [Archiv speichern/versenden] erstellen Sie eine komprimierte Archivdatei (`*.zip`), die alle für ein Projekt relevanten Dateien enthält. Die Archivdatei können Sie im Dateisystem abspeichern oder als E-Mail-Anhang versenden.

Hinweis:

Die Archiv-Funktion ist nicht geeignet, um Projektumgebungen wiederherzustellen. Sie ist nur zur einfachen Zusammenfassung aller projektzugehörigen Dateien gedacht. Beim Entpacken eines Archivs müssen Sie die Verzeichnispfade der einzelnen Dateien an die aktuelle PLC-Editor-Umgebung anpassen!

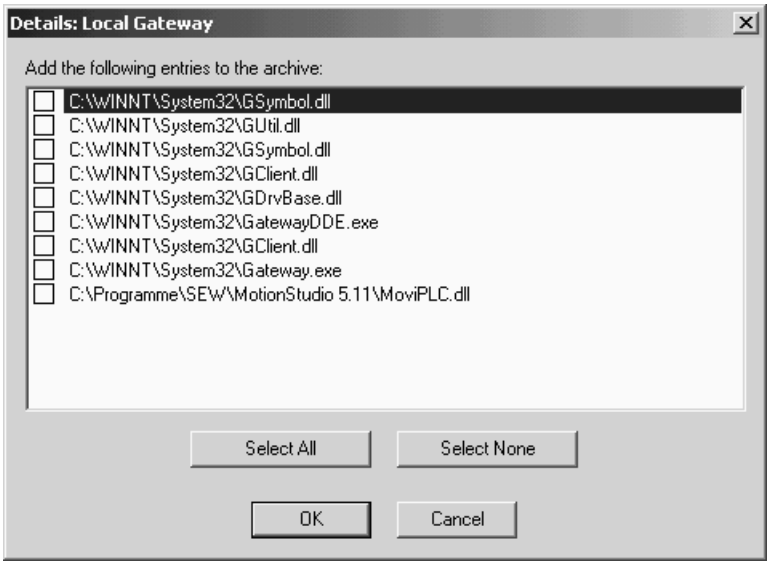
Einstellungen für Archivdatei festlegen

Nach dem Ausführen des Menübefehls wird der Dialog [Archiv speichern] ([Save Archive]) geöffnet:

20279AEN

Wählen Sie die Datei-Kategorien aus, die dem Projektarchiv hinzugefügt werden sollen. Eine Kategorie gilt als ausgewählt, wenn die Kontrollbox davor mit einem Haken versehen ist. Klicken Sie zum An- bzw. Abwählen auf die Kontrollbox. Für jede ausgewählte Kategorie werden standardmäßig alle relevanten Dateien in die Archivdatei kopiert. Für einige Kategorien können Sie jedoch eine Teilauswahl festlegen.

Um den zur Kategorie gehörenden Dialog [Details: <Kategoriename>] ([Details: <Kategoriename>]) zu öffnen, klicken Sie auf die Schaltfläche [Details] ([Details]):



20285AEN

Der Dialog zeigt eine Liste aller in dieser Kategorie verfügbaren Dateien. Aktivieren bzw. deaktivieren Sie die gewünschten Dateien durch einen Klick auf die Kontrollbox. Mit den Schaltflächen [Alles auswählen] und [Nichts auswählen] ([Select All] und [Select None]) erfassen Sie alle Dateien der Liste.

Um die Auswahl zu übernehmen, schließen Sie den Dialog [Details] mit [OK].

Die Einstellung wird bis zur endgültigen Erstellung des Dateiarchivs gespeichert.

Im Hauptdialog [Archiv speichern] erkennen Sie die Kategorien, für die eine Teilauswahl vorgenommen wurde, am grauen Hintergrund der Kontrollbox: .

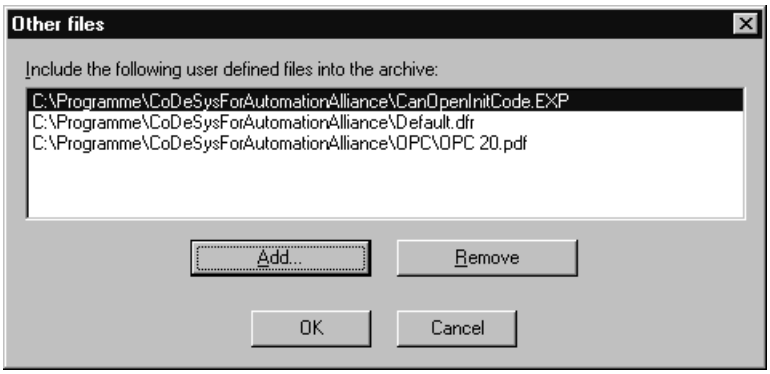
Die folgende Tabelle zeigt, welche Dateikategorien vordefiniert sind und welche Dateitypen sie jeweils automatisch einbeziehen:

Kategorie		Dateierweiterungen	Erläuterung zu den Dateien
[Projektdatei]	[Project File]	<Projektname>.pro	Projektdatei
[Referenzierte Bibliotheken]	[Referenced Libraries]	*.lib, *.obj, *.hex	Bibliotheken und gegebenfalls die zugehörigen Dateien mit den Erweiterungen `*.obj` und `*.hex`
[Compile-Informationen]	[Compile Information]		Information des letzten Übersetzungslaufs, auch für die Simulation
		*.ci	Information des letzten Übersetzungslaufs
		*.ri	Download-Information
		<temp>.*	Temporäre Übersetzungs- und Downloaddateien
[INI-Datei]	[INI File]	*.ini	`PLCEditor.ini`
[Logbuch]	[Log File]	*.log	Projekt-Logbuch
[Registrierungseinträge]	[Registry entries]	registry.reg	Einträge für Gateway und SPS; Folgender Zweig aus der Registry wird einbezogen: `HKEY_LOCAL_MACHINE\SOFTWARE\SEW`
[Symbol-Dateien]	[Symbol Files]	*.sdb, *.sym	Aus dem Projekt erzeugte Symbolinformation
[Konfigurationsdateien]	[Configuration Files]	z. B.: *.cfg, *.con, *.eds, *.dib, *.ico	Dateien für die Steuerungskonfiguration: Konfigurationsdateien, Geräte-Stammdateien, Icons etc.
[Target-Dateien]	[Target Files]	*.trg	Target-Dateien im Binärformat für alle installierten Targets
		*.txt	Target-Dateien im Textformat für alle installierten Targets, wenn verfügbar

Kategorie		Dateierweiterungen	Erläuterung zu den Dateien
[Lokaler Gateway]	[Local Gateway]	Gateway.exe, GatewayDDE.exe, GClient.dll, GDrvBase.dll, GDrvStd.dll, Ghandle.dll, GSymbol.dll, GUtil.dll	Gegebenenfalls weitere im Gateway-Verzeichnis vorhandene Dateien mit der Erweiterung `*.dll`

Um beliebige andere Dateien zur Archivdatei hinzuzufügen, klicken Sie auf die Schaltfläche [Andere Dateien] ([Other Files]).

Der Dialog [Andere Dateien] ([Other files]) wird geöffnet.



20287AEN

In diesem Dialog können Sie eine Liste zusätzlicher Dateien erstellen.

Um eine Datei zur Liste hinzuzufügen, klicken Sie auf die Schaltfläche [Hinzufügen] ([Add]). Der Dialog [Öffnen] ([Open]) wird geöffnet. Wählen Sie die Datei aus und bestätigen Sie mit [Öffnen] ([Open]). Die Datei wird in die Liste im Dialog [Andere Dateien] übernommen.

Um eine Datei aus der Liste zu entfernen, klicken Sie auf die Schaltfläche [Entfernen] ([Remove]).

Wenn die Liste fertig ist, schließen Sie den Dialog mit [OK], um die Einstellungen zu speichern.

Sie können der Archivdatei eine Datei `readme.txt` mit Informationen zum Dateiarchiv hinzuzufügen. Klicken Sie im Dialog [Archiv speichern] ([Save archive]) auf die Schaltfläche [Kommentar] ([Comment]), um den gleichnamigen Dialog zu öffnen. Der Dialog enthält ein Eingabefeld zur Eingabe des Texts.

Wenn Sie den Dialog mit [OK] beenden, wird bei der Erstellung der Archivdatei eine Datei namens `readme.txt` erzeugt. Die Datei `readme.txt` enthält den eingegebenen Text, das Erstellungsdatum und die Versionsnummer des PLC-Editor.

Archivdatei erstellen

Nachdem Sie alle Einstellungen vorgenommen haben, erstellen Sie die Archivdatei.

Folgende Optionen stehen zur Verfügung:

- [Speichern] ([Save])

  Die Option erstellt und speichert die Archivdatei. Der Dialog [Speichern unter] ([Save as]) wird geöffnet. Wählen Sie ein Verzeichnis zum Speichern aus und geben Sie den Dateinamen ein. Der Dateiname ist standardmäßig `<projektname>.zip`.

  Wenn Sie mit [Speichern] bestätigen, wird die Generierung der Archivdatei gestartet. Der Fortschritt der Generierung wird in einem Dialog angezeigt und im Meldungsfenster protokolliert.

- [Senden] ([Mail])

  Die Option erstellt eine temporäre `*.zip`-Datei und generiert eine leere E-Mail, die

die Archivdatei als <projektname>.zip als Anhang enthält. Diese Funktion setzt eine korrekte Installation des MAPI (Messaging Application Programming Interface) voraus. Während die E-Mail erzeugt wird, erscheint ein Fortschrittsdialog und der Ablauf wird im Meldungsfenster protokolliert. Die temporäre *.zip-Datei wird gelöscht, sobald die Archivdatei der E-Mail als Anhang beigefügt ist.

Wenn die E-Mail-Option nicht korrekt funktioniert, wenden Sie sich an Ihren Systemadministrator.

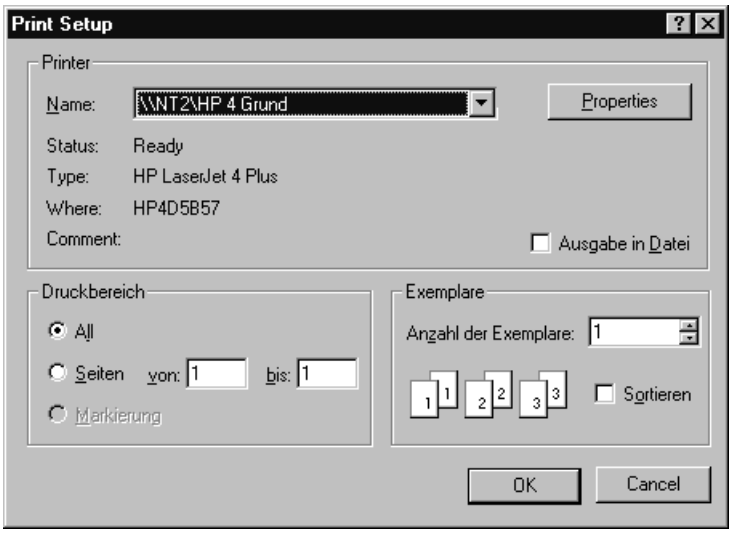
Über die Schaltfläche [Abbrechen] ([Cancel]) brechen Sie den Dialog ohne Erstellung einer Archivdatei ab. Die vorgenommenen Einstellungen werden nicht gespeichert.

4.3.8 [Datei] / [Drucken] ([File] / [Print])

Tastaturbefehl: <Strg>+<P>

Mit dem Menübefehl [Datei] / [Drucken] wird der Inhalt des aktiven Fensters gedruckt.

Nach dem Ausführen des Menübefehls wird der Dialog [Drucken] ([Print Setup]) geöffnet. Nehmen Sie die entsprechenden Einstellungen vor oder konfigurieren Sie den Drucker. Farbausdrucke sind aus allen Editoren möglich.



20288AEN

Um den Dialog zur Druckereinrichtung zu öffnen, klicken Sie auf die Schaltfläche [Eigenschaften] ([Properties]).

Das Layout Ihres Ausdrucks können Sie mit dem Menübefehl [Datei] / [Einstellungen Dokumentation] ([File] / [Printer Setup]) festlegen.

Um die Seitenaufteilung bereits während der Arbeit in den Editorfenstern berücksichtigen zu können, haben Sie Möglichkeit, die Anzeige der aktuellen Druckbereichsgrenzen zu aktiviern. Wählen Sie dazu den Menübefehl [Projekt] / [Optionen], Kategorie [Arbeitsbereich] ([Project] / [Options], Kategorie [Desktop]). Aktivieren Sie in dieser Kategorie die Option [Druckbereiche anzeigen] ([Show print area margins]).

Um den Druck zu starten, bestätigen Sie mit [OK]. Das aktive Fenster wird ausgedruckt.

Während des Druckens wird in einer Dialogbox die Anzahl der bereits gedruckten Seiten angezeigt. Wenn Sie diese Dialogbox schließen, bricht der Druckvorgang nach der nächsten Seite ab.

Um Ihr gesamtes Projekt zu dokumentieren, wählen Sie [Projekt] / [Dokumentieren] ([Project] / [Document]).

Wenn Sie für das Projekt eine Dokumentvorlage mit vorgegebenen Kommentaren für alle Projektvariablen erstellen wollen, öffnen Sie eine globale Variablenliste und führen Sie den Menübefehl [Extras] / [Doku-Vorlage erstellen] ([Extras] / [Make Docuframe file]) aus.

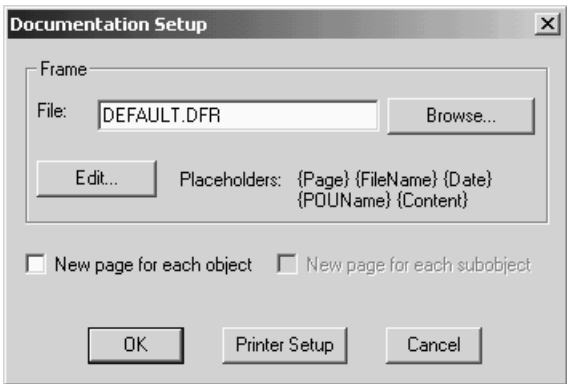
Wenn Sie eine bestehende Dokumentvorlage verwenden wollen, führen Sie den Menübefehl [Extras] / [Doku-Vorlage auswählen] ([Extras] / [Link Docu file]) aus. Der Dialog [Öffnen] ([Open]) zur Auswahl einer Vorlagedatei (`*.txt`) wird geöffnet.

Wenn beim Ausführen des Menübefehls [Datei] / [Drucken] der Fokus im Meldungsfenster ist, wird dessen gesamter Inhalt wie dargestellt ausgedruckt.

4.3.9 [Datei] / [Einstellungen Dokumentation] ([File] / [Printer Setup])

Mit dem Menübefehl [Datei] / [Einstellungen Dokumentation] legen Sie das Layout der ausgedruckten Seiten fest.

Nach dem Ausführen des Menübefehls wird der Dialog [Einstellungen Dokumentation] ([Documentation Setup]) geöffnet:



20289AEN

Im Eingabefeld [Datei] ([File]) können Sie den Verzeichnispfad und den Namen (`*.dfr`) der Datei eintragen, in der das Seitenlayout gespeichert werden soll. Standardmäßig wird die Vorlage in der Datei "default.dfr" angelegt.

Wenn Sie ein vorhandenes Layout verändern möchten, klicken Sie auf die Schaltfläche [Durchsuchen] ([Browse]). Der Dialog [Öffnen] ([Open]) wird geöffnet, in dem Sie die gewünschte Datei auswählen.

Den Seitenumbruch können Sie über die Optionen [Neue Seite je Objekt] und [Neue Seite je Unterobjekt] ([New page for each object] und [New page for each subobject]) steuern.

Mit der Schaltfläche [Einrichtung] ([Printer Setup]) öffnen Sie den Dialog für die Druckereinrichtung.

Wenn Sie auf die Schaltfläche [Bearbeiten] ([Edit]) klicken, wird die Vorlage zur Einstellung des Seitenlayouts geöffnet. Hier können Sie Platzhalter und andere Objekte auf der Seite platzieren und den Textbereich, in den die Dokumentation gedruckt werden soll, festlegen. Die Blattfläche, die durch die Druckereinrichtung vorgegeben wird, ist rot schraffiert.

Einen Platzhalter fügen Sie über den Menübefehl [Einfügen] / [Platzhalter] ([Insert] / [Placeholder]) ein. In diesem Menü wählen Sie einen der fünf Platzhalter aus. Durch Aufziehen eines Rechteckes mit gedrückter linker Maustaste legen Sie Position und Größe des Platzhalters fest. Die Platzhalter werden im Ausdruck durch den eigentlichen Text ersetzt:

Befehl		Platzhalter	Wirkung im Ausdruck
[Page]	[Seite]	{Page}	Aktuelle Seitennummer
[POU-Name]	[Bausteinname]	{POUName}	Name des aktuellen Bausteins
[Filename]	[Dateiname]	{FileName}	Name des Projekts
[Date]	[Datum]	{Date}	Aktuelles Datum
[Content]	[Inhalt]	{Content}	Inhalt des Bausteins

Wenn Sie die das Fenster schließen, werden Sie gefragt, ob Sie die Änderungen speichern wollen.

Hinweis:

Um die später für das Ausdrucken des Projekts vorgesehene Blattgröße bereits während des Programmierens berücksichtigen zu können, stellen Sie das gewünschte Format wie hier beschrieben ein und aktivieren Sie in den Projektoptionen, Kategorie [Arbeitsbereich] ([Desktop]) die Option [Druckbereiche anzeigen] ([Show print area margins]).

Objekte einfügen

In diesem Abschnitt erfahren Sie, wie Sie Visualisierungsobjekte zur näheren Beschreibung des Projekts auf der Seite einfügen können.

Rechteck

Menübefehl: [Einfügen] / [Rechteck] ([Insert] / [Rectangle])

Schaltfläche: [ ]

Vorgehensweise:

1. Führen Sie den Menübefehl [Einfügen] / [Rechteck] aus.
2. Klicken Sie im Seitenlayout-Fenster an die Stelle, an der das Zeichenobjekt eingefügt werden soll und ziehen Sie bei gedrückter Maustaste das Zeichenobjekt auf die gewünschte Größe.

   Sie können nun das Zeichenobjekt weiterbearbeiten und das Aussehen verändern.

Abgerundetes Rechteck

Menübefehl: [Einfügen] / [Abgerundetes Rechteck] ([Insert] / [Rounded Rectangle])

Schaltfläche: [ ]

Vorgehensweise:

1. Führen Sie den Menübefehl [Einfügen] / [Abgerundetes Rechteck] aus.
2. Klicken Sie im Seitenlayout-Fenster an die Stelle, an der das Zeichenobjekt eingefügt werden soll und ziehen Sie bei gedrückter Maustaste das Zeichenobjekt auf die gewünschte Größe.

   Sie können nun das Zeichenobjekt weiterbearbeiten und das Aussehen verändern.

Ellipse

Menübefehl: [Einfügen] / [Ellipse] ([Insert] / [Ellipse])

Schaltflächen: [ ]

Vorgehensweise:

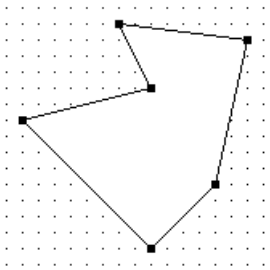
1. Führen Sie den Menübefehl [Einfügen] / [Ellipse] aus.
2. Klicken Sie im Seitenlayout-Fenster an die Stelle, an der das Zeichenobjekt eingefügt werden soll und ziehen Sie bei gedrückter Maustaste das Zeichenobjekt auf die gewünschte Größe.

   Sie können nun das Zeichenobjekt weiterbearbeiten und das Aussehen verändern.

Polygon

Menübefehl: [Einfügen] / [Polygon] ([Insert] / [Polygon])

Schaltfläche: [ ]



20221AXX

Ein Polygon ist ein geschlossener Linienzug, der sich aus mehreren Linien zusammensetzt. Start- und Endpunkt sind identisch.

1. Führen Sie den Menübefehl [Einfügen] / [Polygon] aus.
2. Klicken Sie im Seitenlayout-Fenster an die Stelle, an der das Polygon eingefügt werden soll und ziehen Sie die Maus.

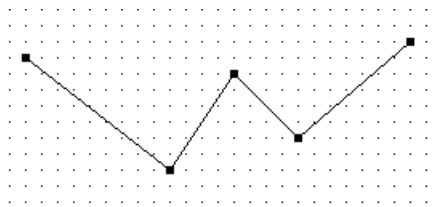
   Es erscheint eine Linie, deren Lage und Länge Sie durch Ziehen mit der Maus verändern.
3. Wenn die Linie die gewünschte Länge und Lage hat, klicken Sie mit der linken Maustaste.
4. Ziehen Sie die Maus, um eine weitere Linie einzufügen.

   Der Linienzug wird automatisch zu einem Polygon ergänzt.
5. Wiederholen Sie die Schritte 3 und 4 für eine beliebige Anzahl weiterer Linien.
6. Schließen Sie die Erstellung des Polygons mit einem Doppelklick ab.

Linienzug

Menübefehl: [Einfügen] / [Linienzug] ([Insert] / [Polyline])

Schaltfläche: [ ]



20222AXX

Ein Linienzug setzt sich aus mehreren, miteinander verbundenen Linien zusammen. Start- und Endpunkt sind nicht identisch.

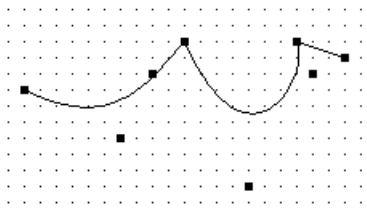
1. Führen Sie den Menübefehl [Einfügen] / [Linienzug] aus.
2. Klicken Sie im Seitenlayout-Fenster an die Stelle, an der der Linienzug eingefügt werden soll und ziehen Sie die Maus.

   Es erscheint eine Linie, deren Lage und Länge Sie durch Ziehen mit der Maus verändern.
3. Wenn die Linie die gewünschte Länge und Lage hat, klicken Sie mit der linken Maustaste.
4. Ziehen Sie die Maus, um eine weitere Linie einzufügen.
5. Wiederholen Sie die Schritte 3 und 4 für eine beliebige Anzahl weiterer Linien.
6. Schließen Sie die Erstellung des Linienzugs mit einem Doppelklick ab.

Kurve

Menübefehl: [Einfügen] / [Kurve] ([Insert] / [Curve])

Schaltfläche: [ ]



20223AXX

Eine Kurve besteht aus einem oder mehreren Bögen. Diese Kurven werden auch als Bezier-Kurven bezeichnet.

1. Führen Sie den Menübefehl [Einfügen] / [Kurve] aus.
2. Klicken Sie im Seitenlayout-Fenster an die Stelle, an der die Kurve eingefügt werden soll und ziehen Sie die Maus.

   Es erscheint eine Linie, deren Lage und Länge Sie durch Ziehen mit der Maus verändern. Der Endpunkt der Linie markiert den Scheitelpunkt der späteren Bogens.
3. Wenn die Linie die gewünschte Länge und Lage hat, klicken Sie mit der linken Maustaste.

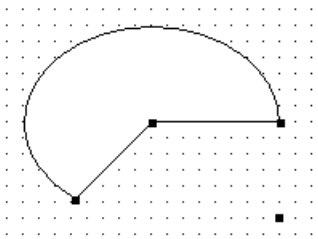
   Es erscheint eine weitere Linie, deren Anfangspunkt mit dem Endpunkt der ersten Linie verbunden ist. Das Ende der zweiten Linie markiert den Endpunkt des Bogens.
4. Ziehen das Ende der zweiten Linie an die gewünchte Position.
5. Wenn Sie nur eine einzelne Kurve erstellen wollen, schließen Sie jetzt mit einem Doppelklick ab. Die beiden geraden Linien werden automatisch in eine Kurve umgewandelt.

   Wenn Sie eine Kurve mit mehreren Bögen erstellen wollen, wiederholen Sie die Schritte 3 und 4 für eine beliebige Anzahl weiterer Bögen. Schließen Sie Erstellung mit einem Doppelklick ab.

Kreissektor

Menübefehl: [Einfügen] / [Kreissektor] ([Insert] / [Pie])

Schaltfläche: [ ]



20224AXX

Ein Kreissektor ist ein Ausschnitt eines Kreises oder einer Ellipse.

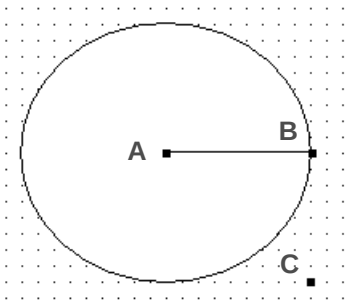
1. Führen Sie den Menübefehl [Einfügen] / [Kreissektor] aus.
2. Klicken Sie im Seitenlayout-Fenster, um die Position des Mittelpunkts festzulegen und ziehen Sie die Maus.

   Es erscheint ein Kreis, dessen Größe und Form Sie durch Ziehen der Maus verändern. Je nachdem, wie Sie die Maus bewegen entsteht ein Kreis oder eine Ellipse.
3. Wenn der Kreis bzw. die Ellipse die gewünschte Größe und Form hat, klicken Sie mit der linken Maustaste.

   Es erscheint ein Vollkreissektor mit einer waagerechten Linie.

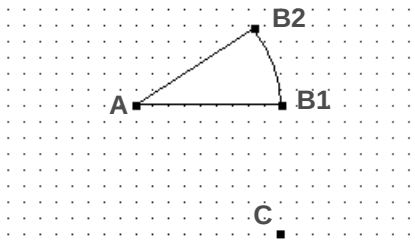
Kreissektor bearbeiten

Sie können ein Kreissektor bearbeiten, indem Sie die Greifpunkte verschieben:



20225AXX

- Um einen Kreissektor mit einem Winkel kleiner als 360° zu erzeugen, klicken Sie auf den Greifpunkt B und verschieben Sie ihn entlang des Kreisbogens (gegen oder in Uhrzeigerrichtung).



20226AXX

- Durch Verschieben des Greifpunkts A verändern Sie die Position des Scheitelpunkts.
- Durch Verschieben der Greifpunkte B1 und B2 verändern Sie Anfangs- und Endwinkel des Kreissektors.

  Hinweis: Sie können die Werte für die beiden Winkel auch direkt eingeben, indem Sie den Menübefehl [Extras] / [Konfiguration] ([Extras] / [Configure]) wählen. Anschließen geben Sie in der Kategorie [Winkel] die Werte für die beiden Winkel ein.
- Durch Verschieben des Greifpunkts C ändern Sie Radius und Bogenlänge des Kreissektors.

Bitmap

Menübefehl: [Einfügen] / [Bitmap] ([Insert] / [Bitmap])

Schaltfläche: [ ]

Sie haben die Möglichkeit, eine Bitmap-Grafik (*.bmp) in das Seitenlayout einzufügen.

1. Führen Sie den Menübefehl [Einfügen] / [Bitmap] aus.
2. Klicken Sie in das Seitenlayout-Fenster, halten Sie die Maustaste und ziehen Sie ein Rechteck auf.
3. Lassen Sie die Maustaste los.

   Es öffnet sich der Dialog [Öffnen].
4. Wählen Sie die gewünschte Grafik aus und drücken Sie auf [Öffnen].

   Die Grafik wird in den Rahmen übernommen.

Platzhalter

In diesem Abschnitt erfahren Sie, wie Sie Platzhalter in das Seitenlayout einfügen. Platzhalter sind eine Abkürzung für wiederkehrende Informationen (wie z. B. aktuelles Datum oder Seitenzahl) und werden im Ausdruck durch die entsprechende Informationen ersetzt.

1. Führen Sie den Menübefehl [Einfügen] / [Platzhalter] ([Insert] / [Placeholder]) und anschließend den Platzhalter, den Sie einfügen wollen.

Es stehen Ihnen fünf Platzhalter zur Verfügung:

- Seite
- Bausteinname
- Dateiname
- Datum
- Inhalt

2. Klicken Sie in das Seitenlayout-Fenster und ziehen Sie bei gedrückter Maustaste ein Rechteck auf.

   An der Stelle erscheint ein schwarz umrahmtes Rechteck mit dem Namen des Platzhalters.

   Wiederholen Sie das Einfügen für die weiteren Platzhalter. Jeder Platzhalter außer [Inhalt] kann auch mehrfach eingefügt werden.

   Hinweis: Wenn im Menü der Platzhalter [Inhalt] nicht auswählbar ist, obwohl im Seitenlayout noch kein Platzhalter [Inhalt] eingefügt ist, hat keiner der Bausteine des Projekts einen Inhalt.

Menü Extras

Nach vorne bringen

Menübefehl: [Extras] / [Nach vorne bringen] ([Extras] / [Bring to front])

Mit dem Menübefehl [Extras] /[Nach vorne bringen] legen Sie die markierten Visualisierungsobjekte in den Vordergrund.

Hinweis:

Mithilfe des Menübefehls [Extras] / [Elementliste] können Sie sämtliche Elemente nach vorne bzw. hinten bringen.

Nach hinten bringen

Menübefehl: [Extras] / [Nach hinten bringen] ([Extras] /[Bring to back])

Mit dem Menübefehl [Extras] /[Nach hinten bringen] legen Sie die markierten Visualisierungsobjekte in den Hintergrund.

Hinweis:

Mithilfe des Menübefehls [Extras] / [Elementliste] können Sie sämtliche Elemente nach vorne bzw. hinten bringen.

Hintergrundbitmap einfügen

Menübefehl: [Extras] / [Hintergrundbitmap auswählen] ([Extras] / [Select background bitmap])

Mit dem Menübefehl [Extras] / [Hintergrundbitmap auswählen] wird ein Bitmap als Hintergrund für das Seitenlayout ausgewählt.

Nach dem Ausführen des Menübefehls wird die Dialog [Öffnen] ([Open]) geöffnet. Wählen Sie im Dateiverzeichnis das Bitmap aus und klicken Sie auf [Öffnen] ([Open]). Das Bitmap wird als Hintergrundbitmap eingefügt.

Im Gegensatz zu den anderen Objekten lässt sich das Hintergrundbitmap nicht markieren. Um das Hintergrundbitmap wieder zu löschen, verwenden Sie den Menübefehl [Extras] / [Hintergrundbitmap löschen] ([Clear background bitmap]).

Hintergrundbitmap löschen

Menübefehl: [Extras] / [Hintergrundbitmap löschen] ([Extras] / [Clear background bitmap])

Mit dem Menübefehl [Extras] / [Hintergrundbitmap löschen] wird das Hintergrundbitmap gelöscht.

Informationen zum Einfügen eines Hintergrundbitmap finden Sie auf Seite 112 (→ Abschnitt. Hintergrundbitmap einfügen).

Objekte ausrichten

Menübefehl: [Extras] / [Ausrichten] ([Extras] / [Align])

Zu dem Menübefehl [Extras] / [Ausrichten] gibt es sechs Untermenübefehle:

[Links]	[Left]	Alle selektierten Objekte werden an dem am weitesten links liegenden Objekt linksbündig ausgerichtet.
[Rechts]	[Right]	Alle selektierten Objekte werden an dem am weitesten rechts liegenden Objekt rechtsbündig ausgerichtet.
[Oben]	[Top]	Alle selektierten Objekte werden an dem am weitesten oben liegenden Objekt an der Oberkante ausgerichtet.
[Unten]	[Bottom]	Alle selektierten Objekte werden an dem am weitesten unten liegenden Objekt an der Unterkante ausgerichtet.
[Horizontal zentrieren]	[Horizontal center]	Alle selektierten Objekte werden in deren gemeinsamen horizontalen Schwerpunkt mittig ausgerichtet.
[Vertikal zentrieren]	[Vertical center]	Alle selektierten Objekte werden in deren gemeinsamen vertikalen Schwerpunkt mittig ausgerichtet.

Mit dem Menübefehl können Sie mehrere Visualisierungsobjekte zueinander ausrichten.

Markieren Sie dazu die Visualisierungsobjekte und führen Sie den gewünschten Untermenübefehl aus.

Alles markieren

Menübefehl: [Extras] / [Alles markieren] ([Extras] / [Select all])

Mit dem Menübefehl [Extras] / [Alles markieren] markieren Sie alle Objekte auf der Seite.

Um nur einen Teil der Objekte zu markieren, gehen Sie folgendermaßen vor:

Wenn Sie sich noch nicht im Selektionsmodus befinden - der Mauszeiger hat in diesem Modus die Form eines weißen Pfeils, klicken Sie in der Funktionsleiste auf die Schaltfläche [ ].

Um ein Objekt zu markieren, klicken Sie mit der linken Maustaste auf das Objekt. Ebenso können Sie durch Drücken der <Tabulator>-Taste das erste Objekt der Elementliste und mit jedem weiteren Drücken das jeweils nächste Objekt anwählen.

Wenn Sie die <Umschalt>-Taste gedrückt halten und anschließend die <Tabulator>-Taste drücken wählen Sie das vorhergehende Objekt an.

Um Objekte zu markieren, die unter einem anderen liegen, markieren Sie mit einem Mausklick zunächst das oberste Objekt. Klicken Sie anschließend bei gedrückter Tastenkombination <Strg>+<Umschalt> mit der Maus um nacheinander die darunter liegenden Objekte auszuwählen.

Um mehrere Objekte zu markieren, halten Sie die <Umschalt>-Taste gedrückt und klicken Sie mit der Maus nacheinander auf die entsprechenden Objekte oder ziehen Sie bei gedrückter linker Maustaste einen Rahmen um die zu markierenden Objekte.

Wenn Sie sich in der Elementliste befinden, können Sie auch von dort durch Anwahl einer Zeile ein Objekt markieren.

Selektieren

Menübefehl: [Extras] / [Selektieren] ([Extras] / [Select mode])

Tastaturbefehl: <Strg>+ rechte Maustaste

Schaltfläche: [ ]

Mit dem Menübefehl [Extras] / [Selektieren] schalten Sie den Selektionsmodus ein und aus.

Konfigurieren

Menübefehl: [Extras] / [Konfigurieren] ([Extras] / [Configure])

In diesem Abschnitt wird beschrieben, wie Sie Einzelobjekte konfigurieren. Informationen zum Konfigurieren von Objektgruppen finden Sie auf Seite 119 (→ Abschnitt. Gruppierung konfigurieren).

Mit dem Menübefehl [Extras] / [Konfigurieren] öffnen Sie für das markierte Objekt den Dialog [Element konfigurieren] ([Regular Element Configuration]). Anzahl und Aufbaud der Kategorien des Dialogs unterscheideen sich je nach markiertem Objekt.

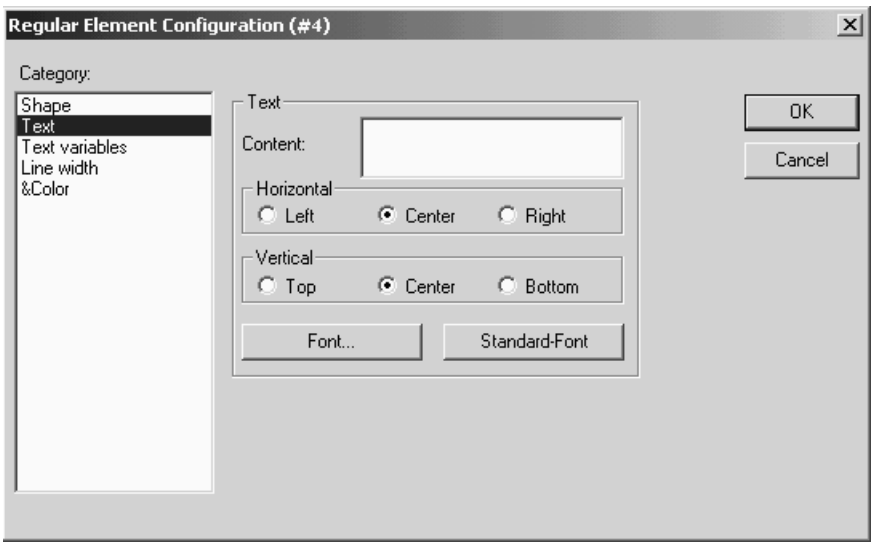
Folgende Kategorien sind für alle Objekttypen gleich:

- Text
- Textvariablen
- Linienstärke

Folgende Kategorien unterscheiden sich nach Objekttyp:

- Form
- Textvariablen
- Farben
- Winkel

Kategorie [Text] ([Text])

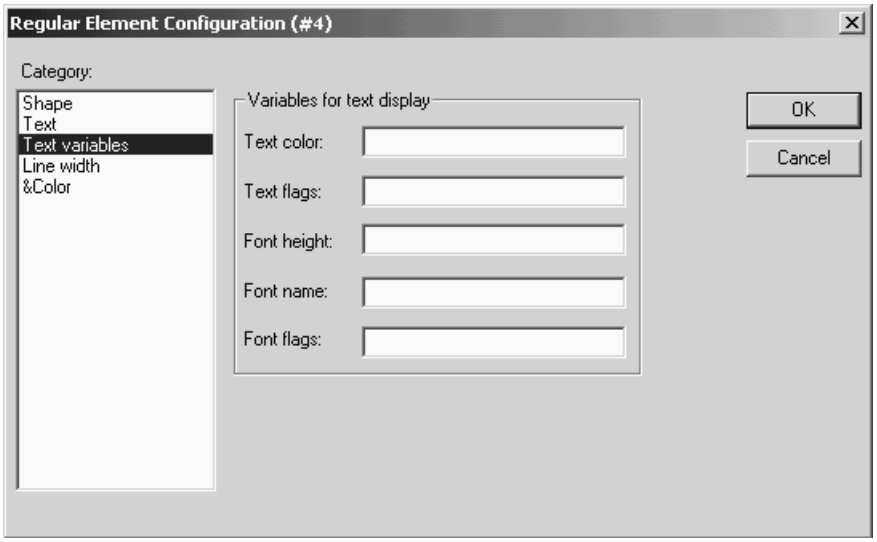


20702AEN

In der Kategorie [Text] können Sie eine Beschriftung für das Objekt einfügen. Im Eingabefeld [Inhalt] ([Content]) geben Sie den Text ein, der beim Objekt erscheinen soll. Mit den Optionen [Horizontal] und [Vertikal] ([Horizontal] und [Vertical]) legen Sie die Ausrichtung des Texts in Bezug auf das Objekt fest.

Außerdem können Sie über die Schaltfläche [Schriftart] ([Font]) die Einstellungen für die Schrift ändern. Wenn Sie auf die Schaltfäche [Standard-Schrift] ([Standard-Font]) klicken, werden die Schrifteinstellungen wieder auf die in den Projektoptionen, Kategorie [Editor] ([Editor]), festgelegten Einstellungen zurückgesetzt.

Kategorie [Textvariablen] ([Text variables])

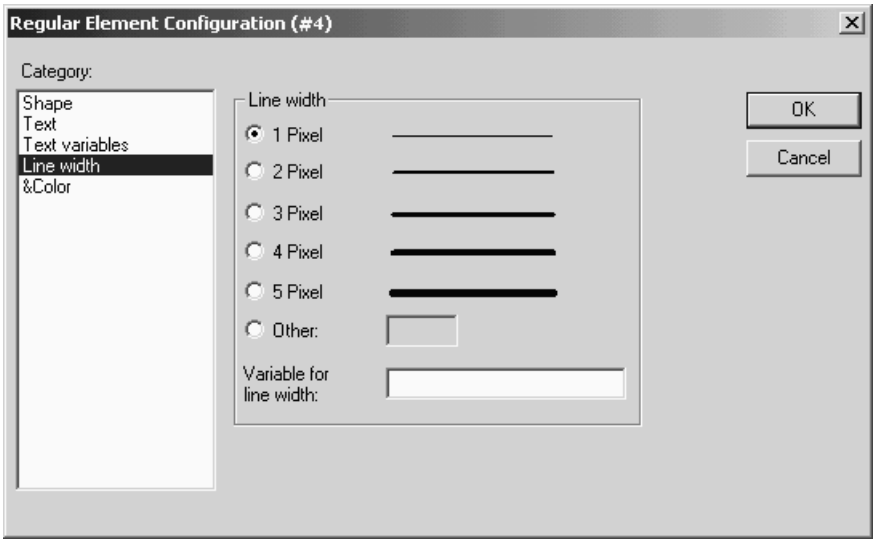


20703AEN

In der Kategorie [Textvariablen] können Sie folgende Textvariablen festlegen:

- [Textfarbe] ([Text color])
- [Textflags] ([Text flags])
- [Fonthöhe] ([Font height])
- [Fontname] ([Font name])
- [Fontflags] ([Font flags])

Kategorie [Linienstärke] ([Line width])



20704AEN

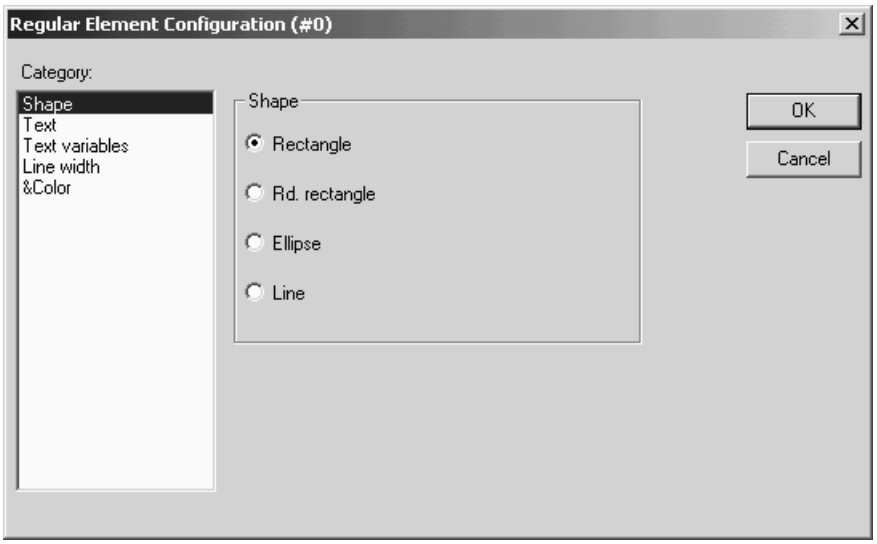
In der Kategorie [Linienstärke] können Sie die Linienstärke für das markierte Objekt verändern oder weisen dem Objekt eine Variable zu.

Kategorie [Form] ([Form])

In der Kategorie [Form] können Sie die Grundform des Objekts ändern. Beim Öffnen des Dialogs ist immer die aktuelle Form des Objekts markiert.

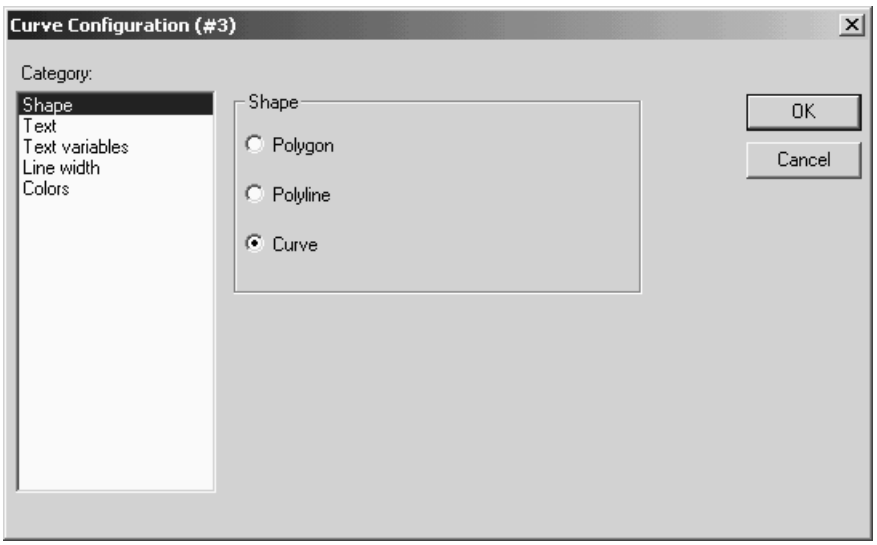
Es gibt zwei Versionen dieser Kategorie, abhängig vom Objekttyp.

Rechteck, abgerundetes Rechteck, Ellipse, Linie



20220ADE

Polygon, Linienzug, Kurve



20706AEN

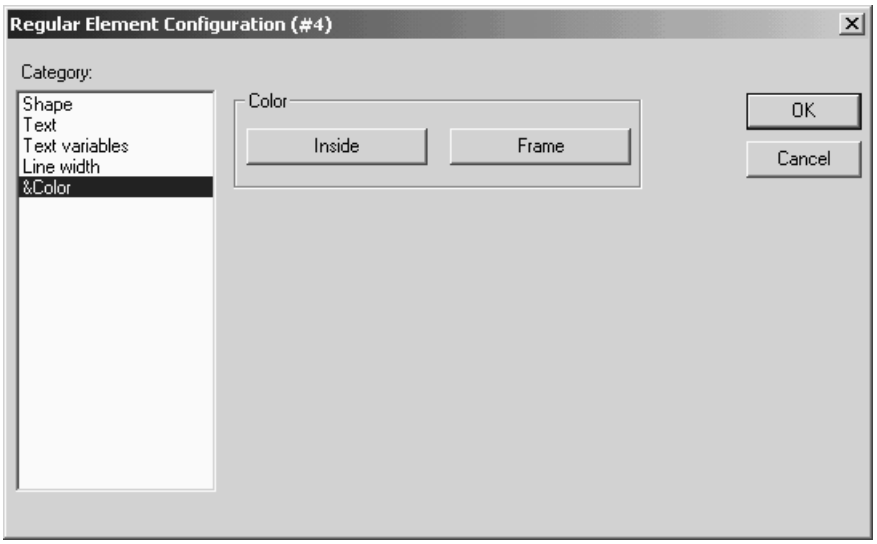
Für den Objekttyp **Kreissektor** ist diese Kategorie nicht verfügbar.

Kategorie [Farbe] ([Color])

In der Kategorie [Farbe] können Sie die Farbeinstellungen des markierten Objekts ändern.

Es gibt zwei Versionen dieser Kategorie, abhängig vom Objekttyp.

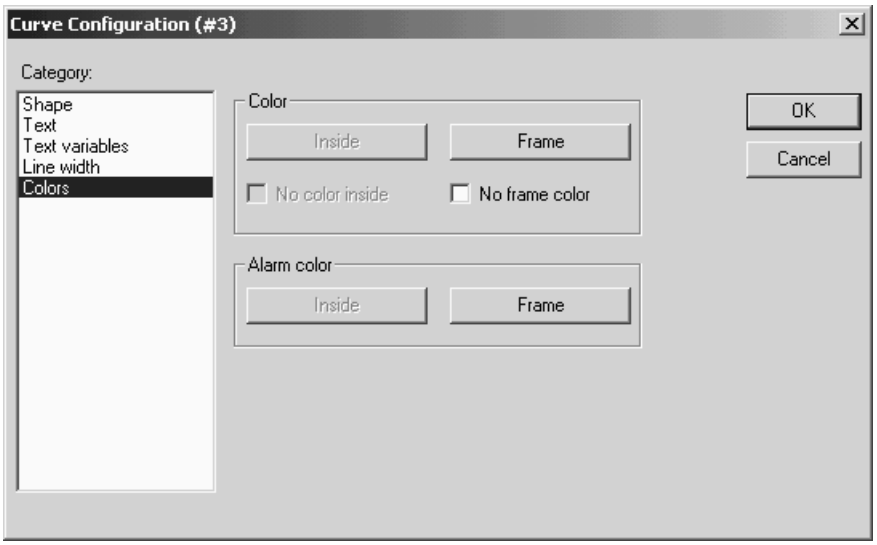
Rechteck, abgerundetes Rechteck, Ellipse, Linie



20705AEN

Für diese Objekttypen können Sie die Innenfarbe und die Rahmenfarbe festlegen.

Polygon, Linienzug, Kurve, Kreissektor

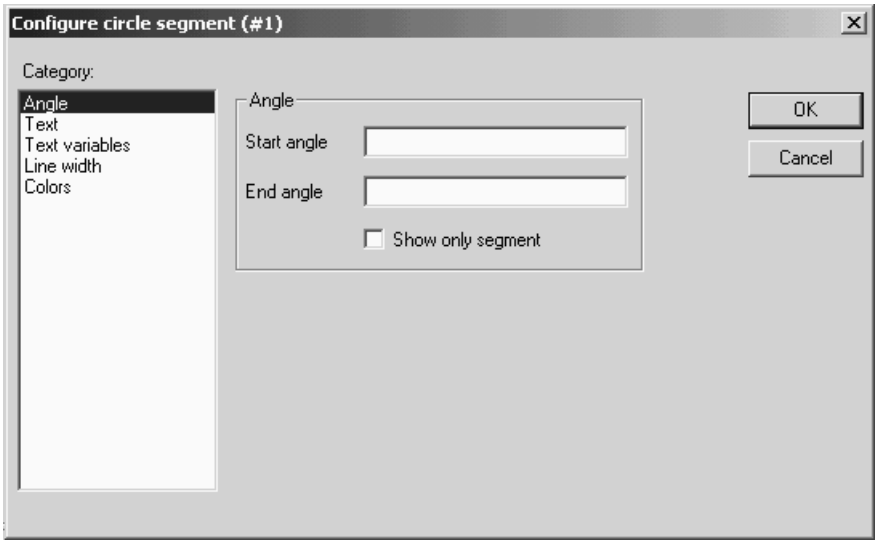


20708AEN

Für diese Objekttypen können Sie festlegen, ob die Objekte überhaupt eine Innenfarbe oder eine Rahmenfarbe haben sollen und anschließend die Farben festlegen.

Außerdem haben Sie bei diesen Objekten die Möglichkeit, Alarmfarben festzulegen.

Kategorie [Winkel] ([Angle])



20707AEN

Die Kategorie [Winkel] steht nur für den Objekttyp **Kreissektor** zur Verfügung. Sie können Anfangs- und Endwinkel für den Kreissektor festlegen. Wenn Sie die Option [nur Kreisbogen anzeigen] ([Show only segment]) aktivieren, wird nur der Kreisbogen angezeigt und nicht der gesamte Kreissektor.

Gruppieren

Menübefehl: [Extras] / [Gruppieren] ([Extras] / [Group])

Mit dem Menübefehl [Extras] / [Gruppieren] können Sie Objekte gruppieren und anschließend Aktionen für alle Objekte der Gruppe gemeinsam ausführen.

Um eine Gruppe zu bilden, markieren Sie die Objekte, die in der Gruppe zusammengefasst werden sollen.

Um ein Objekt zu markieren, klicken Sie mit der linken Maustaste auf das Objekt.

Um mehrere Objekte zu markieren, halten Sie die <Umschalt>-Taste gedrückt und klicken Sie mit der Maus nacheinander auf die entsprechenden Objekte oder ziehen Sie bei gedrückter linker Maustaste einen Rahmen um die zu markierenden Objekte.

Um Objekte zu markieren, die unter einem anderen liegen, markieren Sie mit einem Mausklick zunächst das oberste Objekt. Klicken Sie anschließend bei gedrückter Tastenkombination <Strg>+<Umschalt> mit der Maus um nacheinander die darunter liegenden Objekte auszuwählen.

Nachdem Gruppieren werden die Objekte der Gruppe bei weiteren Aktionen wie folgt behandelt:

- Die Objekte erhalten einen gemeinsamen Rahmen. Beim Ziehen an den Eckpunkten des Rahmens werden alle Objekte gleich verzerrt. Das Verschieben ist nur in der Gruppe möglich.
- Alle Objekte erhalten gemeinsame Eigenschaften. Änderungen in den Eigenschaften betreffen deshalb immer die ganze Gruppe. Weitere Informationen zum Konfigurieren von Gruppen finden Sie im nächsten Abschnitt.

Mit dem Menübefehl [Extras] / [Gruppierung aufheben] ([Extras] / [Ungroup]) heben Sie die Gruppierung wieder auf. Dabei gehen alle Einstellungen der Gruppenkonfiguration verloren.

Gruppierung konfigurieren

Eine Gruppe von Objekten kann konfiguriert werden. Markieren Sie dazu die Gruppe und wählen Sie den Menübefehl [Extras] / [Konfigurieren] ([Extras] / [Configure]).

Nach dem Ausführen des Menübefehls wird der Dialog [Gruppierung konfigurieren] ([Group konfiguration]) geöffnet.

Die Einstellungen beziehen sich dann jeweils auf alle Objekte der Gruppe. Welche Einstellungen möglich sind, hängt von den Objekttttypen in der Gruppe ab

Der Dialog enthält folgende Kategorien:

- [Gruppierung] ([Group])
- [Text] ([Text])
- [Textvariablen] ([Text variables])
- [Farbvariablen] ([Color variables])
- [Linienstärke] ([Line width])
- [Bewegung absolut] ([Motion absolute])
- [Bewegung relativ] ([Motion relative])
- [Variablen] ([Variables])
- [Eingabe] ([Input])
- [Text für Tooltip] ([Text for Tooltip])
- [Zugriffsrechte] ([Security])
- [Programmierfähigkeit] ([Programmability])

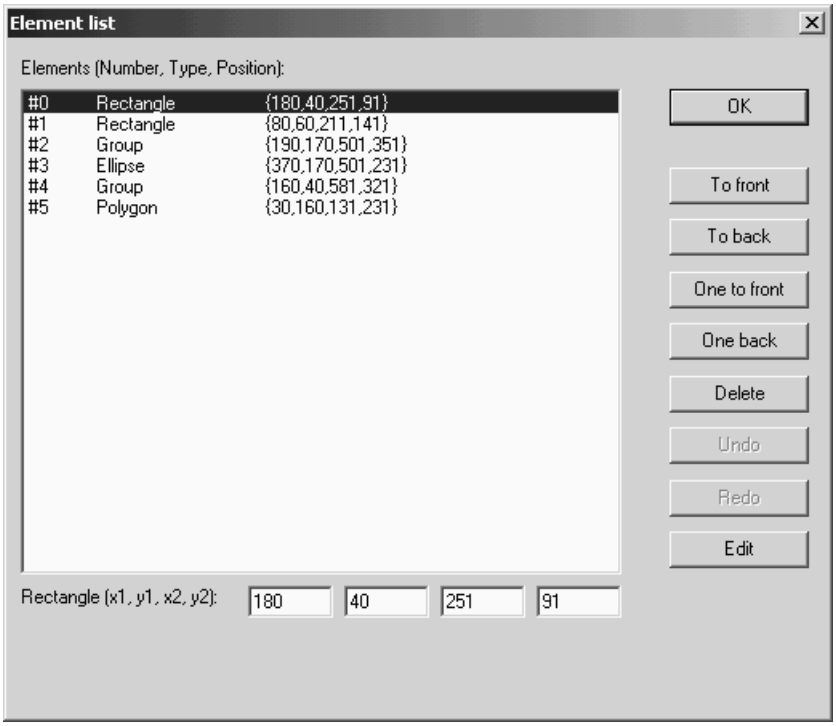
Gruppierung aufheben

Menübefehl: [Extras] / [Gruppierung aufheben] ([Extras] / [Ungroup])

Mit dem Menübefehl [Extras] / [Gruppierung aufheben] heben Sie die Gruppierung für die markierte Objektgruppe wieder auf. Dabei gehen alle Gruppeneinstellungen verloren, die Sie über den Menübefehl [Extras] / [Konfigurieren] ([Extras] / [Configure]) vorgenommen haben.

Elementliste

Menübefehl: [Extras] / [Elementliste] ([Extras] / [Elementlist])



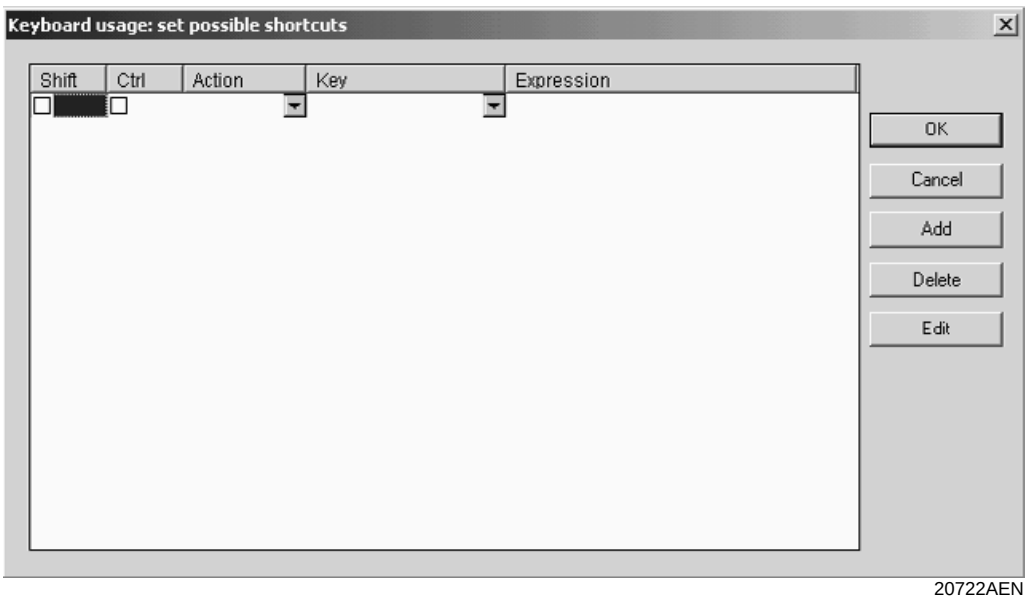
20721AEN

Nach dem Ausführen des Menübefehls [Extras] / [Elementliste] wird der Dialog [Elementliste] ([Element list]) geöffnet.

Der Dialog enthält eine Liste aller Objekte, die zum Seitenlayout gehören. In dem Dialog können Sie Objekte in den Vorder- bzw. Hintergrund stellen oder Objekte löschen. Außerdem können Sie die Koordinaten der Objektpunkte bearbeiten.

Tastaturbedienung

Menübefehl: [Extras] / [Tastaturbedienung] ([Extras] / [Keyboard usage])



20722AEN

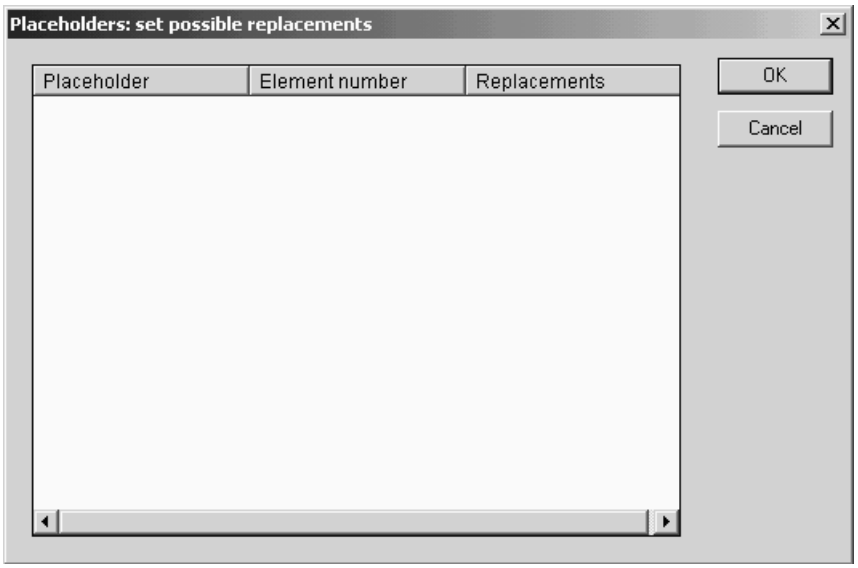
In dem Dialog [Tastaturbedienung] haben Sie die Möglichkeit, für Aktionen eigene Tastaturbefehle festzulegen.

Wenn Sie die Tasten <Umschalt> und/oder <Strg> verwenden möchten, aktivieren Sie die Optionen [Shift] und/oder [Ctrl]. Zusammen mit der unter [Taste] ([Key]) festgelegten

Taste bilden die Tasten <Umschalt> und/oder <Strg>. Wenn Sie
Wählen Sie anschließend die Aktion, für die der Tastaturbefehl gelten soll.

Platzhalterliste

Menübefehl: [Extras] / [Platzhalterliste] ([Extras] / [List of placeholders])



20723AEN

In dem Dialog werden alle vorhandenen Platzhalter aufgelistet.

4.3.10 [Datei] / [Beenden] ([File] / [Exit])

Tastaturbefehl: <Alt>+<F4>

Mit dem Menübefehl [Datei] / [Beenden] beenden Sie den PLC-Editor.

Wenn ein Projekt geöffnet ist, wird es geschlossen.

Wenn das Projekt nach dem letzten Speichern verändert wurde, erscheint ein Fenster mit der Meldung: "Das Projekt wurde verändert. Vor dem Schließen speichern?"

Wenn das zu speichernde Projekt den Namen "Unbenannt" trägt, müssen Sie einen anderen Namen für das Projekt festlegen.

Weitere Informationen zum Speichern von Projekten finden Sie auf Seite 102 (→ Kap. 4.3.6, [Datei] / [Speichern unter] ([File] / [Save as])).

4.3.11 [Projekt] / [Übersetzen] ([Project] / [Build])

Tastaturbefehl: <F11>

Mit dem Menübefehl [Projekt] / [Übersetzen] wird das Projekt übersetzt (kompiliert). Der Übersetzungsvorgang ist grundsätzlich inkrementell. Nur veränderte und neue Bausteine werden übersetzt. Ein nicht inkrementeller Übersetzungsvorgang ist auch mit diesem Befehl möglich, wenn Sie vorher den Menübefehl [Projekt] / [Alles bereinigen] ([Project] / [Clean all]) ausführen.

Bei Zielsystemen, die **Online Change** unterstützen, sind nach dem Übersetzungslauf im **Object Organizer** alle Bausteine, die beim nächsten Download auf die Steuerung geladen werden, mit einem blauen Pfeil gekennzeichnet.

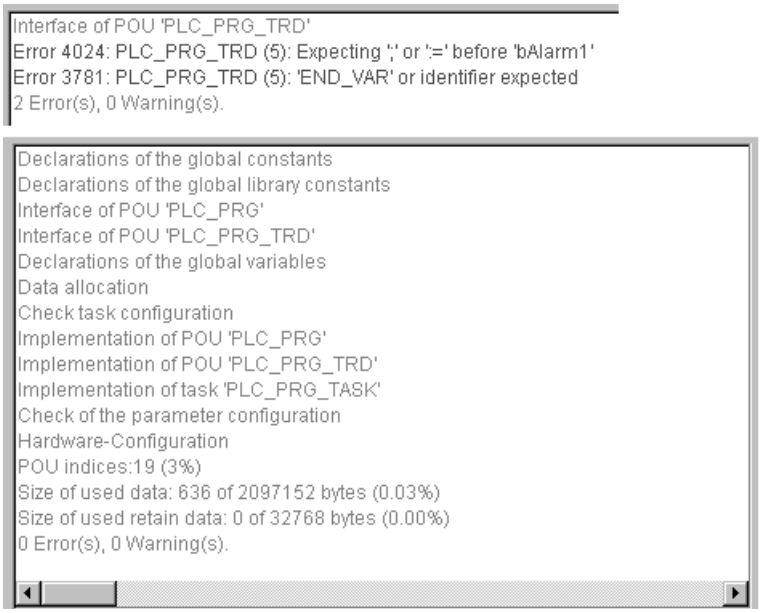
Der Übersetzungslauf wird automatisch ausgeführt, wenn Sie sich über den Menübefehl [Online] / [Einloggen] ([Online] / [Login]) in die Steuerung einloggen.

Beim Übersetzen wird das Meldungsfenster geöffnet, in dem folgende Informationen angezeigt werden:

- Fortschreiten des Übersetzungsvorgangs
- Fehler und Warnungen (mit Nummern)
- Angaben zu Indizes bzw. Speicherverbrauch (jeweils Anzahl und Prozentsatz)

Um Informationen zu einem Fehler zu erhalten, markieren Sie den Fehler und drücken Sie die <F1>-Taste.

Beispiel für Fehlermeldungen und Übersetzungsinformationen im Meldungsfenster eines Projekts


Interface of POU 'PLC_PRG_TRD'
Error 4024: PLC_PRG_TRD (5): Expecting ';' or '=' before 'bAlarm1'
Error 3781: PLC_PRG_TRD (5): 'END_VAR' or identifier expected
2 Error(s), 0 Warning(s).
Declarations of the global constants
Declarations of the global library constants
Interface of POU 'PLC_PRG'
Interface of POU 'PLC_PRG_TRD'
Declarations of the global variables
Data allocation
Check task configuration
Implementation of POU 'PLC_PRG'
Implementation of POU 'PLC_PRG_TRD'
Implementation of task 'PLC_PRG_TASK'
Check of the parameter configuration
Hardware-Configuration
POU indices:19 (3%)
Size of used data: 636 of 2097152 bytes (0.03%)
Size of used retain data: 0 of 32768 bytes (0.00%)
0 Error(s), 0 Warning(s).

20727AEN

Wenn in den Projektoptionen, Kategorie [Laden & Speichern] ([Load & Save]), die Option [Automatisch sichern vor Übersetzen] ([Auto save before compile]) aktiviert ist, wird das Projekt vor dem Übersetzen gespeichert.

Sie können Objekte vom Übersetzen ausschließen:

- im **Object Organizer** über das Kontextmenü, Menübefehl [Vom Übersetzen ausschließen] ([Exclude from build])
- in den Projektoptionen, Kategorie [Übersetzungsoptionen] ([Build]) über die Schaltfläche [Objekte ausschließen] ([Exclude objects])

Hinweis:

Querverweise entstehen während der Übersetzung und werden in den Übersetzungsinformationen mit gespeichert. Um die Befehle aus dem Menü [Projekt] / [Überprüfen] ([Project] / [Check]) anwenden zu können bzw. aktuelle Resultate zu erhalten, müssen Sie das Projekt nach einer Veränderung neu übersetzen.

4.3.12 [Projekt] / [Alles übersetzen] ([Project] / [Rebuild all])

Mit dem Menübefehl [Projekt] / [Alles übersetzen] wird, im Gegensatz zum inkrementellen Übersetzen beim Menübefehl [Projekt] / [Übersetzen] ([Project] / [Build]), das Projekt komplett neu übersetzt. Im Gegensatz zum Menübefehl [Project] / [Alles bereinigen] ([Project] / [Clean all]) werden dabei die Download-Informationen nicht gelöscht.

Sie haben die Möglichkeit, Objekte vom Übersetzen auszuschließen. Informationen dazu finden Sie auf Seite 122 (→ Kap. 4.3.11, [Projekt] / [Übersetzen] ([Project] / [Build])).

4.3.13 [Projekt] / [Alles bereinigen] ([Project] / [Clean all)]

Mit dem Menübefehl [Projekt] / [Alles bereinigen] werden die Informationen des letzten Downloads und des letzten Übersetzungsvorgangs gelöscht.

Eine Dialogbox wird geöffnet mit dem Hinweis, dass **Online Change** nicht mehr möglich ist. Hier können Sie den Befehl abbrechen oder bestätigen.

Hinweis:

Ein Online Change ist auch nach dem Menübefehl [Alles bereinigen] möglich, wenn Sie vorher die Datei *.ri mit den Projektinformationen des letzten Downloads umbenennen oder in ein anderes Dateiverzeichnis verschieben. Vor dem Einloggen können Sie die Download-Informationen wieder laden.

Weitere Informationen finden Sie auf Seite 124 (→ Kap. 4.3.14, [Projekt] / [Download-Information laden] ([Project] / [Load download information])).

4.3.14 [Projekt] / [Download-Information laden] ([Project] / [Load download information])

Mit dem Menübefehl [Projekt] / [Download-Information laden] können Sie die Download-Informationen aus einer `*.ri`-Datei laden.

Nach dem Ausführen des Menübefehls wird der Dialog [Öffnen] ([Open]) geöffnet.

Die Download-Informationen werden bei jedem Download automatisch in einer Datei gespeichert. Bei einigen Zielsystemen geschieht dies außerdem bei jeder Erzeugung eines Bootprojekts im Offline-Modus. Die Datei hat den Namen `<Projektname><Targetidentifier>.ri` und wird im Projektverzeichnis abgelegt.

Die Datei wird bei jedem Öffnen des Projekts geladen und dient beim erneuten Einloggen auf die Steuerung dem ID-Check. Beim ID-Check wird überprüft, ob das geöffnete Projekt dem auf der Steuerung entspricht. Außerdem wird geprüft, bei welchen Bausteinen sich der generierte Code verändert hat. Bei Zielsystemen, die **Online Change** unterstützen, werden beim Download nur diese geänderten Bausteine geladen. Die `*.ri`-Datei ist somit Voraussetzung für einen Online Change.

Hinweis:

Mit dem Menübefehl [Projekt] / [Alles bereinigen] ([Project] / [Clean all]) wird die zum Projekt gehörende `*.ri`-Datei automatisch aus dem Projektverzeichnis gelöscht, so dass zunächst kein Online Change mehr möglich ist. Um das Löschen der Datei zu verhindern, benennen Sie die `*.ri`-Datei vor dem Menübefehl um oder verschieben Sie die Datei in ein anderes Dateiverzeichnis. Anschließend können Sie die Download-Informationen wieder aus der ursprünglichen `*.ri`-Datei laden.

4.3.15 [Projekt] / [In andere Sprache übersetzen] ([Project] / [Translate into other language])

Der Menübefehl [Projekt] / [In andere Sprache übersetzen] dient dazu, die aktuelle Projektdatei in eine andere Sprache zu übersetzen bzw. in einer anderen Sprache darzustellen. Dies geschieht durch das Einlesen einer Übersetzungsdatei, die aus dem Projekt erzeugt und mit Hilfe eines Texteditors mit Übersetzungstexten ergänzt wurde.

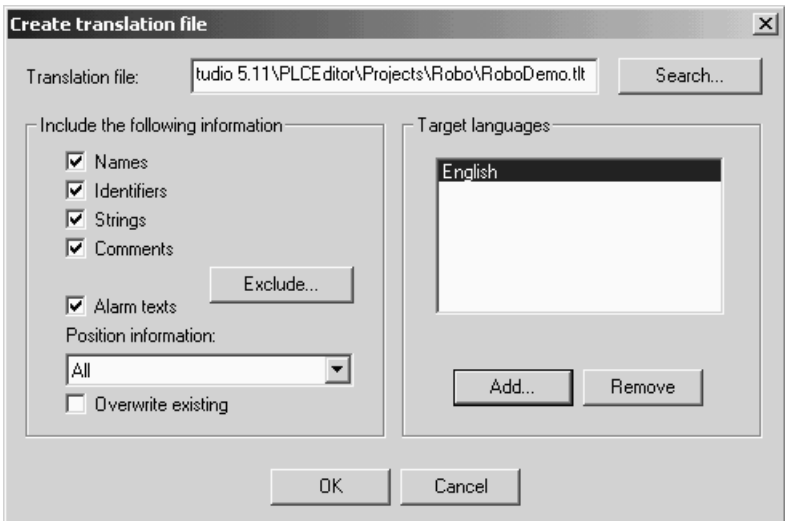
Der Menübefehl hat folgende Untermenüpunkte:

- [Übersetzungsdatei erstellen] ([Create translation file])
- [Projekt übersetzen] ([Translate this project])
- [Projekt übersetzt darstellen] ([View translated project])
- [Übersetzung umschalten] ([Toggle translation])

Weitere Informationen finden Sie auf Seite 127 (→ Abschnitt. Übersetzungsdatei bearbeiten).

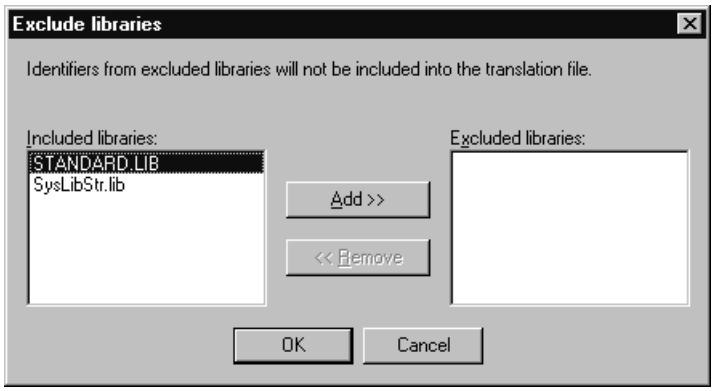
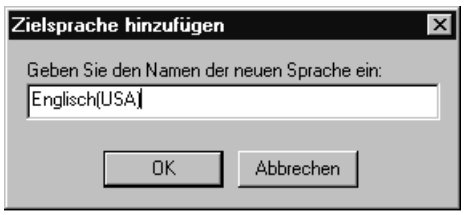
Übersetzungsdatei erstellen

Nach dem Auswählen des Menübefehls [Projekt] / [In andere Sprache übersetzen] / [Übersetzungsdatei erstellen] ([Project] / [Translate into other language] / [Create translation file]) wird der Dialog [Übersetzungsdatei erstellen] ([Create translation file]) geöffnet.



20728AEN

Option		Beschreibung
[Übersetzungsdatei]	[Translation file]	Geben Sie im Eingabefeld [Übersetzungsdatei] Verzeichnispfad und Namen der neuen Übersetzungsdatei ein. Standardmäßig ist die Dateierweiterung `.tlt`. Die Übersetzungsdatei ist eine Textdatei. Wenn Sie die Datei beispielsweise in MS Excel oder MS Word bearbeiten möchten, verwenden Sie am besten die Dateierweiterung `.txt`. Dadurch werden die Daten automatisch in Tabellenform angeordnet. Wenn Sie eine existierende Übersetzungsdatei bearbeiten wollen, wählen Sie die Übersetzungsdatei über die Schaltfläche [Durchsuchen] ([Search]) aus.
[Folgende Informationen aufnehmen]	**[Include the following information]**	In diesem Bereich legen Sie fest, welche Projektinformationen in die Übersetzungsdatei aufgenommen werden sollen: • [Names] ([Names]), z. B. der Titel "Bausteine" im **Object Organizer** • [Identifiers] ([Identifiers]) • [Strings] ([Strings]) • [Kommentare] ([Comment]) • [Alarmtexte] ([Alarm texts])

Option		Beschreibung
[Positionsinformationen]	[Position information]	Die Positionsinformation beschreibt mit Dateipfad, Baustein und Zeile die Position des Sprachsymbols, das zur Übersetzung bereitgestellt wird. Zur Auswahl stehen drei Optionen: • [Keine] ([None]): Es werden keine Positionsinformationen generiert. • [Erstmaliges Auftreten] ([First occurrence]): Die Positionsinformationen des ersten Auftretens des Sprachsymbols werden generiert. • [Alle] ([All]): Zu jedem Auftreten des Sprachsymbols werden Positionsinformationen generiert. Wenn Sie eine existierende Übersetzungsdatei bearbeiten, werden die dort enthaltenen Projektinformationen der betroffenen Sprachsymbole entsprechend den hier gemachten Einstellungen überschrieben. **Hinweis**: Pro Sprachsymbol sind maximal 64 Positionsinformationen möglich.
[Bestehende überschreiben]	[Overwrite existing]	Wenn Sie eine existierende Übersetzungsdatei bearbeiten, werden sämtliche Positionsinformationen übrschrieben, gleichgültig von welchem Projekt sie generiert wurden.
[Ausschließen]	[Exclude]	Der Dialog [Bibliotheken ausschließen] ([Exclude libraries]) wird geöffnet, in dem die in das Projekt eingebundenen Bibliotheken aufgelistet sind.  Um eine Bibliothek aus der Übersetzungsdatei auszuschließen, markieren Sie die Bibliothek in der Liste [Eingebundene Bibliotheken] ([Included libraries]) und klicken Sie anschließend auf die Schaltfläche [Hinzufügen] ([Add]). Um eine Bibliothek in die Übersetzungsdatei einzubinden, markieren Sie die Bibliothek in der Liste [Ausgeschlossene Bibliotheken] ([Excluded libraries]) und klicken Sie anschließend auf die Schaltfläche [Entfernen] ([Remove]). Um die Einstellungen übernommen werden, klicken Sie zum Abschluss auf [OK].
[Zielsprachen]	**[Target languages]**	Die Liste enthält alle Sprachen, die in der Übersetzungsdatei enthalten sind bzw. die nach Beenden des Dialogs [Übersetzungsdatei erstellen] aufgenommen werden sollen. Wenn Sie auf die Schaltfläche [Hinzufügen] klicken, wird der Dialog [Zielsprache hinzufügen] ([Add Target Language]) geöffnet.  Geben Sie einen Namen für die Sprache ein. Der Name darf keine Leerzeichen enthalten. Durch Klicken auf [OK] schließen Sie den Dialog und der Name erscheint in der Liste der Zielsprachen. Um einen Eintrag aus der Liste zu entfernen, markieren Sie den Eintrag und klicken Sie auf [Entfernen] ([Remove]).

Nachdem Sie alle Einstellungen für die Übersetzungsdatei vorgenommen haben, klicken Sie auf [OK]. Dadurch wird der Dialog geschlossen und die Generierung der Über-

setzungsdatei begonnen. Wenn bereits eine gleichnamige Übersetzungsdatei existiert, erscheint folgender Hinweis:

"Die angegebene Übersetzungsdatei existiert bereits. Sie wird nun entsprechend geändert, wobei eine Sicherungskopie der bereits bestehenden Datei angelegt wird. Möchten Sie fortfahren?"

Mit [Nein] ([No]) kehren Sie zum Dialog [Übersetzungsdatei erstellen] zurück.

Wenn Sie auf [Ja] ([Yes]) klicken, wird eine Kopie der bereits bestehenden Übersetzungsdatei mit dem Dateinamen `Backup_of_<Übersetzungsdatei>.xlt` im gleichen Verzeichnis angelegt und die Übersetzungsdatei gemäß der eingestellten Optionen modifiziert.

Beim Erzeugen einer Übersetzungsdatei geschieht Folgendes:

- Für jedes Sprachsymbol wird ein Platzhalter (`##TODO`) generiert.

  Weitere Informationen hierzu finden Sie auf Seite 127 (→ Abschnitt. Übersetzungsdatei bearbeiten).
- Wenn Sie eine existierende Übersetzungsdatei verändern, werden Sprachen, die aus der Liste der Zielsprachen entfernt wurden, in der Übersetzungsdatei ebenfalls entfernt. Das gilt auch für Sprachen, die in anderen Projekten generiert wurden.

Übersetzungsdatei bearbeiten

Sie müssen die Übersetzungsdatei als Textdatei öffnen und speichern. Die Zeichen "##" kennzeichnen Schlüsselwörter. Die Platzhalter `##TODO` in der Datei können durch die gültigen Übersetzungstexte ersetzt werden. Pro Sprachsymbol wird ein durch Typenkennzeichner begrenzter Abschnitt angelegt.

Zum Beispiel kennzeichnen `##NAME_ITEM` und `##END_NAME_ITEM` Start bzw. Ende des Abschnitts für einen Objektnamen im **Object Organizer**. `COMMENT_ITEM` kennzeichnet Abschnitte für Kommentare, `IDENTIFIER_ITEM` Abschnitte für Bezeichner und `STRING_ITEM` Abschnitte für Strings.

Achten Sie darauf, dass übersetzte Bezeichner und Namen gemäß der Norm gültig bleiben und dass Strings und Kommentare in die entsprechenden Klammerzeichen eingeschlossen werden.

Beispiele: Bei einem Kommentar (`##COMMENT_ITEM`), der mit `(* Kommentar 1 *)` in der Übersetzungsdatei steht, muss der Platzhalter `##TODO` durch ein `(* comment 1 *)` ersetzt werden. Bei einem String (`##STRING_ITEM`) muss `'zeichenfolge1'` durch `'string1'` ersetzt werden.

Vorsicht:

Folgende Teile der Übersetzungsdatei sollten Sie nicht bearbeiten, wenn Ihnen die genaue Kenntnisse fehlen: Sprachblock, Flagblock, Positionsinformationen, Originaltexte.

Projekt übersetzen

Das aktuelle Projekt kann unter Verwendung einer gültigen Übersetzungsdatei in eine andere Sprache übersetzt werden.

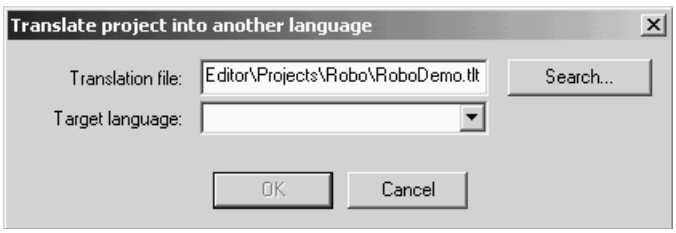
Hinweis:

Ein Übersetzungslauf kann nicht rückgängig gemacht werden.

Wenn Sie die ursprüngliche Sprachversion des Projekts erhalten möchten, speichern Sie vor dem Übersetzen eine Kopie des Projekts.

Wenn Sie das Projekt nur in einer anderen Sprache betrachten möchten, ohne es zu bearbeiten, verwenden Sie den Menübefehl [Projekt] / [In andere Sprache übersetzen] / [Projekt übersetzt darstellen] ([Project] / [Translate into other languages] / [View translated project]).

Nach dem Ausführen des Menübefehls [Projekt] / [In andere Sprache übersetzen] / [Projekt übersetzen] ([Project] / [Translate into other languages] / [Translate this project]) wird der Dialog [Projekt in andere Sprache übersetzen] ([Translate project into another language]) geöffnet.



20731AEN

Geben Sie im Eingabefeld [Übersetzungsdatei] ([Translation file]) Verzeichnispfad und Name der zu verwendenden Übersetzungsdatei an. Alternativ können Sie über die Schaltfläche [Durchsuchen] ([Search]) die Datei im Verzeichnis direkt auswählen.

Anschließend wählen Sie im Auswahlfeld [Zielsprache] ([Target language]) die gewünschte Zielsprache aus.

Um den Übersetzungsprozess zu starten, schließen Sie den Dialog mit [OK], . Während der Übersetzung werden ein Fortschrittsdialog und gegebenenfalls Fehlermeldungen angezeigt. Nach der Übersetzung werden die Dialogbox sowie alle geöffneten Editierfenster geschlossen.

Falls die Übersetzungsdatei fehlerhafte Eingaben enthält, wird nach dem Klicken auf [OK] eine Fehlermeldung ausgegeben, die Dateipfad und fehlerhafte Zeile ausgibt.

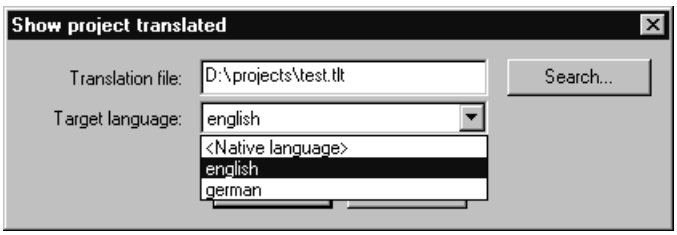
Projekt übersetzt darstellen

Wenn für das Projekt eine Übersetzungsdatei existiert, kann eine der übersetzten Versionen dargestellt werden, ohne dass das Projekt in der Original-Sprachversion überschrieben wird. Sie können das Projekt in dieser Darstellung nicht bearbeiten.

Hinweis:

Für ein wirkliches Übersetzen des Projekts verwenden Sie den Menübefehl [Projekt] / [In andere Sprache übersetzen] / [Projekt übersetzen] ([Project] / [Translate into other languages] / [Translate this project]).

Nach dem Ausführen des Menübefehls [Projekt] / [In andere Sprache übersetzen] / [Projekt übersetzt darstellen] ([Project] / [Translate into other languages] / [View translated project]) wird der Dialog [Projekt übersetzt anzeigen] ([Show project translated]) geöffnet.



20732AEN

Geben Sie im Eingabefeld [Übersetzungsdatei] ([Translation file]) Verzeichnispfad und Name der zu verwendenden Übersetzungsdatei an. Alternativ können Sie über die Schaltfläche [Durchsuchen] ([Search]) die Datei im Verzeichnis direkt auswählen.

Anschließend wählen Sie im Auswahlfeld [Zielsprache] ([Target language]) die gewünschte Zielsprache aus.

Um das Projekt in der gewählten Sprache darzustellen, schließen Sie den Dialog mit [OK].

Um zwischen Originalsprache und Zielsprache zu wechseln, verwenden Sie den Menübefehl [Projekt] / [In andere Sprache übersetzen] / [Übersetzung umschalten] ([Project] / [Translate into other languages] / [Toggle translation]) oder den Tastaturbefehl <Strg>+<Q>.

Übersetzung umschalten

Mit dem Menübefehl [Projekt] / [In andere Sprache übersetzen] / [Übersetzung umschalten] ([Project] / [Translate into other languages] / [Toggle translation]) wechseln Sie zwischen der Originalsprache des Projekts und der Zielsprache.

Die Zielsprache stellen Sie über den Menübefehl [Projekt] / [In andere Sprache übersetzen] / [Projekt übersetzt darstellen] ([Project] / [Translate into other languages] / [View translated project]) ein. Statt des Menübefehls können Sie auch den Tastaturbefehl <Strg>+<Q> benutzen.

4.3.16 [Projekt] / [Dokumentieren] ([Project] / [Document])

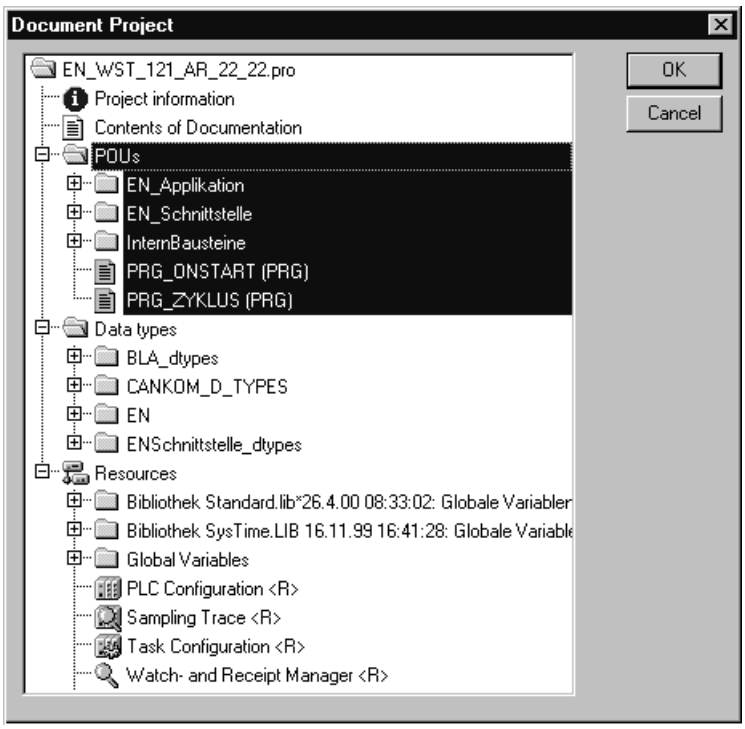
Über den Menübefehl [Projekt] / [Dokumentieren] drucken Sie die Dokumentation des Projekts.

Zu einer vollständigen Dokumentation gehören:

- Bausteine
- Inhaltsübersicht der Dokumentation
- Datentypen
- Ressourcen
  - globale Variablen
  - Variablenkonfiguration
  - Traceaufzeichnung
  - Steuerungskonfiguration
  - Taskkonfiguration
  - Watch- und Rezepturverwalter
- Aufrufbäume von Bausteinen und Datentypen
- Querverweisliste

Für die letzten beiden Punkte muss das Projekt fehlerfrei übersetzt worden sein.

Nach dem Ausführen des Menübefehls wird der Dialog [Projekt dokumentieren] ([Document Project]) geöffnet.



20733AEN

In dem Dialog [Projekt dokumentieren] legen Sie fest, welche Informationen gedruckt werden sollen.

Wenn Sie auf ein Ordnersymbol klicken, wird der gesamte untergeordnete Bereich markiert.

Um die Auswahl auf Teilbereiche oder einzelne Objekte zu beschränken, verwenden Sie die Tasten <Umschalt> und <Strg> plus Mausklick.

Beenden Sie den Dialog mit [OK]. Der Dialog [Drucken] ([Print]) wird geöffnet, in dem Sie die Druckoptionen festlegen.

Das Layout der Druckseiten legen Sie über den Menübefehl [Datei] / [Einstellungen Dokumentation] ([File] / [Printer Setup]) fest.

4.3.17 [Projekt] / [Exportieren] ([Project] / [Export])

Der PLC-Editor bietet die Möglichkeit, Objekte zu exportieren bzw. zu importieren. Damit können Sie Programme zwischen verschiedenen IEC-Programmiersystemen austauschen.

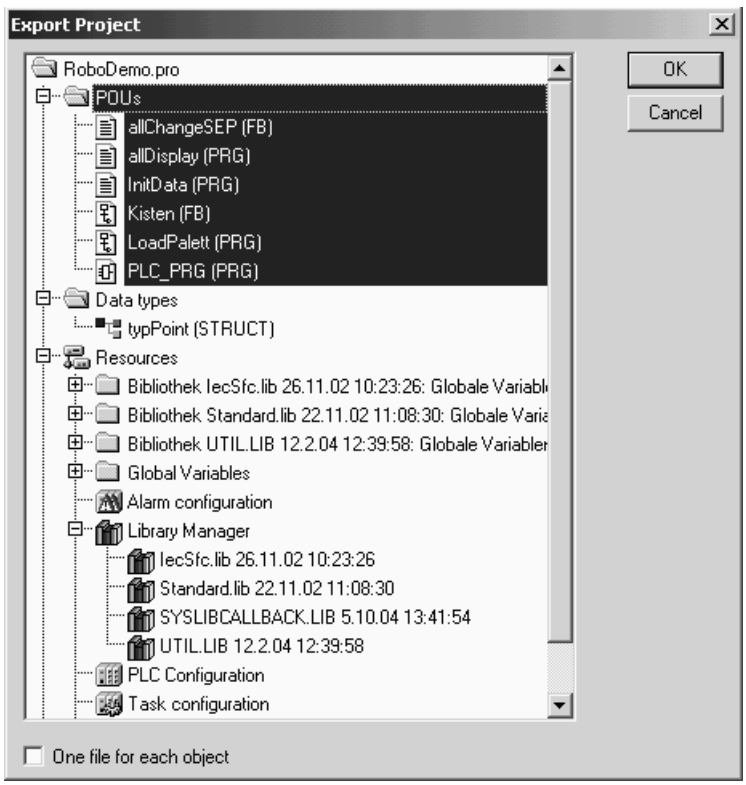
Bisher gibt es ein standardisiertes Austauschformat für Bausteine in AWL, ST und AS: Common Elements der IEC 61131-3.

Für die Bausteine in KOP und FUP und die anderen Objekte hat der PLC-Editor ein eigenes Ablageformat, da es hierfür kein Textformat in der IEC 61131-3 gibt.

Folgende Objekte können exportiert werden:

- Bausteine
- Datentypen
- Ressourcen
- Einträge im Bibliotheksverwalter (nur Verknüpfungsinformationen zu Bibliotheken, nicht die Bibliotheken selbst)

Nach dem Ausführen des Menübefehls [Projekt] / [Exportieren] wird der Dialog [Projekt exportieren] ([Export Project]) geöffnet.



20734AEN

Im Dialog [Projekt exportieren] legen Sie fest, welche Objekte exportiert werden sollen.

Wenn Sie auf ein Ordnersymbol klicken, wird der gesamte untergeordnete Bereich markiert.

Um die Auswahl auf Teilbereiche oder einzelne Objekte zu beschränken, benutzen Sie die Tasten <Umschalt> und <Strg> plus Mausklick.

Nachdem Sie die Objekte ausgewählt haben, können Sie festlegen, ob beim Export eine ASCII-Datei für alle Objekte angelegt wird oder für jedes Objekt eine separate Datei.

Wenn Sie die Option [Eine Datei je Objekt] ([One file for each object]) aktivieren, wird für jedes markierte Objekt eine separate Datei `<Objektname>.exp` angelegt.

Beenden Sie den Dialog mit [OK].

Wenn Sie in eine einzige Datei exportieren, wird der Dialog [Projekt exportieren] ([Export Project]) geöffnet, in dem Sie Verzeichnispfad und Name für die Exportdatei festlegen.

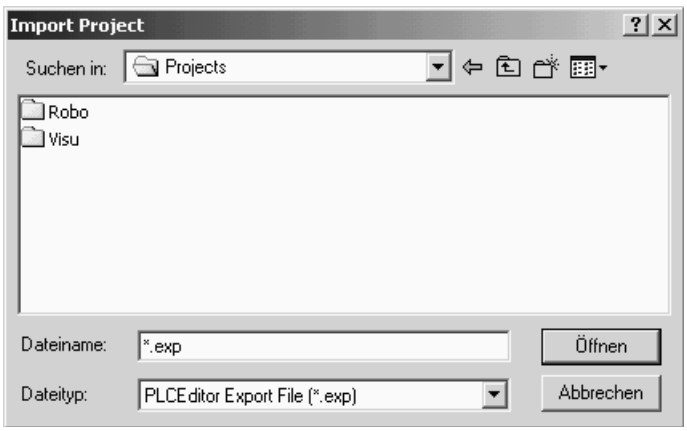
Wenn Sie jedes Objekt in eine separate Datei exportieren, können Sie in dem Dialog nur den Verzeichnispfad festlegen.

Hinweis:

Der Re-Import eines exportierten FUP- oder KOP-Bausteins schlägt fehl, wenn ein Kommentar im grafischen Editor ein Hochkomma (') enthält, da dieses als String-Beginn interpretiert wird!

4.3.18 [Projekt] / [Importieren] ([Project] / [Import])

Nach dem Ausführen des Menübefehls [Projekt] / [Importieren] wird der Dialog [Projekt importieren] ([Import Project]) geöffnet.



20735AEN

Wählen Sie die `*.exp`-Dateien aus, die Sie importieren möchten. Mit den Tasten <Strg> und <Umschalt> plus Mausklick können Sie mehrere Dateien gleichzeitig markieren. Beenden Sie den Dialog mit [OK].

Die Daten werden in das aktuelle Projekt importiert.

Wenn ein gleichnamiges Objekt im Projekt bereits besteht, erscheint ein Hinweis mit der Meldung: "Das Objekt existiert bereits. Wollen Sie es ersetzen?"

Wenn Sie auf [Ja] ([Yes]) klicken, wird das Objekt im Projekt durch das Objekt aus der Importdatei ersetzt. Wenn Sie auf [Nein] ([No]) klicken, erhält der Name des neuen Objekts als Ergänzung einen Unterstrich und eine Zählnummer (z. B. _1). Mit [Ja, alle] bzw. [Nein, alle] ([Yes, all] bzw. [No, all]) wird diese Aktion für alle Objekte bzw. für kein Objekt durchgeführt.

Wenn die Information zur Verknüpfung mit einer Bibliothek importiert wird, wird die Bibliothek geladen und im Bibliotheksverwalter am Ende der Liste angefügt. Wenn die Bibliothek bereits im Projekt enthalten ist, wird sie nicht erneut geladen.

Wenn allerdings in der importierten Exportdatei ein anderer Speicherzeitpunkt für die Bibliothek angegeben ist als in der vorhandenen, wird der Bibliotheksnamen im Bibliotheksverwalter mit einem Sternchen "*" gekennzeichnet
(z. B. `standard.lib*30.3.99 11:30:14`).

Wenn die Bibliothek nicht gefunden werden kann, erscheint der Hinweis: "Kann Biblio-

thek {<Pfad>\}<name> <date> <time> nicht finden".

Der Import wird im Meldungsfenster protokolliert.

4.3.19 [Projekt] / [Siemens Import] ([Project] / [Siemens Import])

Unter dem Menübefehl [Projekt] / [Siemens Import] finden Sie zwei Untermenübefehle zum Import von Bausteinen und Variablen aus Siemens-STEP5-Dateien:

- [SEQ-Symbolikdatei importieren] ([Import an SEQ symbol file])
- [S5-Datei importieren] ([Import an S5 project file])

Der Menübefehl [SEQ-Symbolikdatei importieren] dient zum Import von globalen Variablen aus Step5-Symbolikdateien. Diesen Menübefehl sollten Sie vor dem Menübefehl [S5-Datei importieren] ausgeführen, damit beim Import von Bausteinen aus absoluten Adressen lesbare symbolische Namen erzeugt werden können.

Der Menübefehl [S5-Datei importieren] dient dem Import von Bausteinen aus STEP5-Programmdateien. Dabei werden die Bausteine in das geöffnete Projekt eingefügt. Sie können bestimmen, ob die Bausteine in der Sprache STEP5-AWL bleiben oder in eine IEC-Sprache konvertiert werden.

Hinweis:

Das Projekt, in das Sie importieren, sollte leer sein. Ausnahme: Die Bibliothek `standard.lib` muss im Projekt eingebunden sein, damit Zähler und Timer importiert werden können.

SEQ-Symbolikdatei importieren

Das SEQ-Format ist ein gängiges Format für die Symbolikdatei in einem STEP5-Projekt. Aus SEQ-Symbolik-Dateien (`*.seq`) können symbolische Zuordnungen gelesen werden. Eine symbolische Zuordnung enthält eine absolute Adresse eines S5-Programmelements (Eingang, Ausgang, Merker etc.), einen zugehörigen symbolischen Bezeichner und optional einen Kommentar zum Symbol. Eine SEQ-Datei ist eine Textdatei, die pro Zeile eine solche Zuordnung enthält, wobei die einzelnen Abschnitte der Zuordnung durch Tabulatoren getrennt sind.

Eine Zeile kann auch nur aus einem Kommentar bestehen. Die Zeile muss dann mit einem Semikolon (;) beginnen.

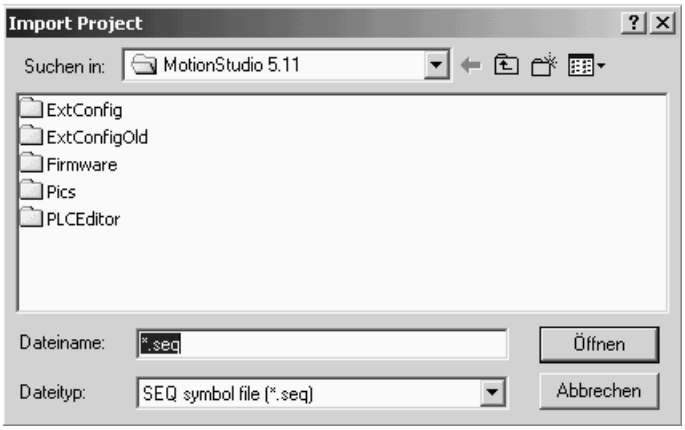
Die symbolischen Zuordnungen in der SEQ-Datei werden in globale Variablendeklarationen nach IEC 1131-3 übersetzt. Dabei werden der symbolische Name, die Adresse und der Kommentar übernommen. Die Adresse wird der IEC 1131-3 angepasst (Prozentzeichen etc.).

Da ein S5-Symbolikname Zeichen enthalten kann, die innerhalb eines IEC-Bezeichners unzulässig sind, wird ggf. der Name geändert. Ungültige Zeichen werden zunächst durch Unterstriche ersetzt; falls mehrere Unterstriche direkt hintereinander stehen würden, wird jeweils der zweite Unterstrich durch ein gültiges Zeichen (z. B. "0") ersetzt. Wenn ein symbolischer Name bei der Konvertierung verändert wurde, wird der Originalname in einem Kommentar dahinter eingefügt. SEQ-Kommentarzeilen werden als Kommentare übernommen. Es können mehrere Blöcke von globalen Variablen erzeugt werden. Jeder Block umfasst weniger als 64 kB Text.

Das beschriebene SEQ-Format wird von der Siemens-STEP5-PG und von ACCON-PG von DELTALOGIC verwendet.

Nach dem Ausführen des Menübefehls [Projekt] / [Siemens Import] / [SEQ-Symboldatei importieren] ([Project] / [Siemens Import] / [Import an SEQ symbol file]) wird der Dialog

[Projekt importieren] ([Import Project]) geöffnet:



20736AEN

Markieren Sie die SEQ-Datei und klicken Sie auf [Öffnen] ([Open]). Der Import wird durchgeführt und abschließend die globale Variablenliste übersetzt.

Hinweis:

Beim Import einer S5-Symbolikdatei können Fehler auftreten, die bedingt sind durch die Umwandlung der STEP5-Bezeichner in IEC-1131-3-konforme Bezeichner bedingt sind. Zum Beispiel werden die STEP5-Bezeichner "A!" und "A?" beide in den IEC-Bezeichner "A_" umgeformt, sodass die Meldung erscheint "Mehrere Deklarationen mit dem gleichen Bezeichner A_". Ändern Sie einen der beiden Bezeichner ab.

Nehmen Sie auf gar keinen Fall sonstige Änderungen an der globalen Variablenliste vor.

Falls Sie Adressen sehen, die auf einer Siemens-SPS gültig, auf Ihrer Steuerung jedoch ungültig sind, ändern Sie die Adressen vorerst nicht ab, auch wenn Sie beim Übersetzen Fehlermeldungen erhalten. Die Original-Adressen werden beim Import der Bausteine gebraucht!

Falls das Projekt, in das Sie importieren, bereits eine Deklaration einer globalen Variable x mit Adresse enthält, kann es sein, dass beim SEQ-Import eine weitere Variable mit derselben Adresse definiert wird. Das ist nach IEC 1131-3 zulässig, aber meistens nicht im Sinne des Anwenders. Sie erhalten keine Fehlermeldungen, aber Ihr Programm wird eventuell nicht wie gewünscht funktionieren, da die Adresse in verschiedenen Bausteinen ohne Zusammenhang benutzt wird. Importieren Sie deshalb das Projekt am besten in ein leeres Projekt, oder in ein Projekt, in dem (noch) keine absoluten Adressen verwendet werden.

Nach dem SEQ-Import können Sie STEP5-Bausteine importieren. Die verwendeten Ein- und Ausgänge müssen beim Import der Bausteine noch nicht in der Steuerungskonfiguration enthalten sein. Spätestens vor dem nächsten Übersetzen des Projekts müssen die Ein- und Ausgänge jedoch eingefügt sein, da beim Übersetzen die verwendeten Adressen geprüft und gegebenenfalls als Fehler gemeldet werden.

S5-Datei importieren

Mit dem Menübefehl [Projekt] / [Siemens Import] / [S5-Datei importieren] ([Project] / [Siemens Import] / [Import an S5 project file]) können Bausteine aus Siemens-S5-Dateien (`*.s5d`) gelesen werden.

Der enthaltene Code ist ein MC5-Code. Der MC5-Code entspricht im Allgemeinen direkt der STEP5-Anweisungsliste (ohne symbolische Namen). Außerdem enthält die `*.s5d`-Datei die Zeilenkommentare der STEP5-Anweisungsliste.

Da eine `*.s5d`-Datei keine symbolischen Namen enthält, sondern nur absolute Adres-

sen, sucht der PLC-Editor den symbolischen Namen für die jeweilige Adresse in den bereits vorhandenen Variablen des PLC-Editor-Projekts. Falls kein symbolischer Name gefunden wird, bleibt die absolute Adresse stehen. Wenn Sie symbolische Namen verwenden möchten, importieren Sie die SEQ-Datei vor der `*.s5d`-Datei.

Nach dem Ausführen des Menübefehls [Projekt] / [Siemens Import] / [S5-Datei importieren] wird der Dialog [Projekt importieren] ([Import Project]) geöffnet. Wählen Sie die gewünschte `*.s5d`-Datei aus und klicken Sie auf [Öffnen] ([Open]). Ein weiterer Dialog wird geöffnet, in dem die enthaltenen Bausteine aufgelistet sind.

Markieren Sie die Bausteine, die importiert werden sollen. Am besten wählen Sie alle Bausteine aus. Sie können hier ebenfalls festlegen, ob die Bausteine in der Sprache STEP5-AWL belassen oder nach AWL, KOP oder FUP konvertiert werden sollen.

Soweit möglich, werden beim Import symbolische Namen anstatt absoluter Adressen verwendet. Wenn der PLC-Editor beim Import eine Anweisung wie `U M12.0` findet, wird nach einer globalen Variablen gesucht, die auf den Merker `M12.0` gelegt ist. Die erste passende Deklaration wird genommen und die Anweisung als `U -Name` importiert, nicht als `U M12.0` importiert. `Name` ist dabei der symbolische Bezeichner für den Merker `M12.0`.

Manchmal werden beim Import bzw. bei der Code-Konvertierung zusätzliche Variablen benötigt. Diese werden global deklariert. Zum Beispiel sind zur Nachbildung flankengetriggerter Eingänge eines S5-Zählers `R_TRIG`-Instanzen nötig.

4.3.20 [Projekt] / [Vergleichen] ([Project] / [Compare])

Der Menübefehl [Projekt] / [Vergleichen] wird verwendet, um zwei Projekte zu vergleichen oder die aktuelle Version des geöffneten Projekts mit der Version, die zuletzt gespeichert wurde.

Definitionen	Beschreibung
aktuelles Projekt	Projekt, das momentan in Bearbeitung ist.
Vergleichsprojekt	Projekt, das zum Vergleich aufgerufen wird.
Vergleichsmodus	In diesem Modus wird das Projekt nach Anwahl des Befehls dargestellt.
Einheit	Kleinste Vergleichseinheit, die aus einer Zeile (Deklarationseditor, ST, AWL), einem Netzwerk (FUP, KOP) oder einem Element/Baustein (CFC, SFC) bestehen kann.

Im Vergleichsmodus werden das aktuelle Projekt und das Vergleichsprojekt in einem zweigeteilten Fenster gegenüber gestellt. Die als unterschiedlich erkannten Bausteine werden farblich gekennzeichnet.

Bei Editorbausteinen gibt es auch für die Inhalte eine direkte Gegenüberstellung.

Vor dem **Vergleichslauf** können Sie Filter bezüglich der Berücksichtigung von Leerzeichen und Kommentaren aktivieren.

Außerdem können Sie wählen, ob im **Vergleichsmodus** Veränderungen innerhalb bestehenbleibender Einheiten als Veränderungen dargestellt oder ob alle unterschiedlichen Einheiten als "neu eingefügt" bzw. "nicht mehr vorhanden" markiert werden.

Die Version des Vergleichsprojekts kann für einzelne Einheiten oder für einen Block markierter Einheiten in das aktuelle Projekt übernommen werden.

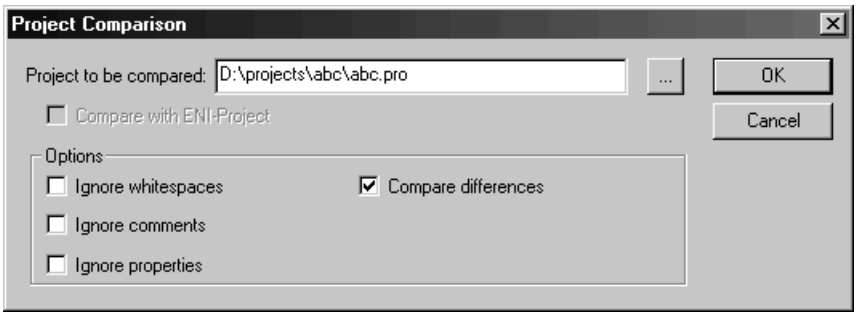
Hinweis:

Solange der Vergleichsmodus aktiviert ist (Meldung in der Statuszeile: COMPARE), können Sie das Projekt nicht editieren!

Weiterführende Informationen finden Sie auf Seite 136 (→ Abschnitt. Projekte vergleichen).

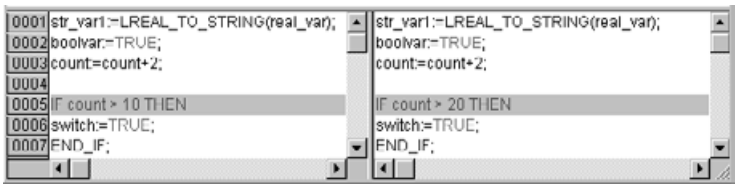
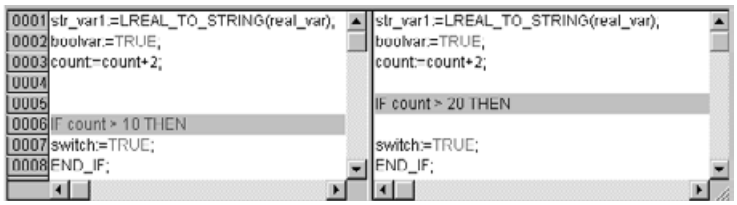
Projekte vergleichen

Nach dem Ausführen des Menübefehls [Projekt] / [Vergleich] ([Project] / [Compare]) wird der Dialog [Projektvergleich] ([Project Comparison]) geöffnet:



20737AEN

Geben Sie im Eingabefeld [Vergleichsprojekt] ([Project to be compared]) Verzeichnispfad und Name des Vergleichsprojekts ein. Alternativ können Sie über die Schaltfläche [...] ein Projekt im Dateiverzeichnis auswählen. Wenn Sie den Namen des aktuellen Projekts eingeben, wird die momentane Fassung des Projekts mit der der letzten Speicherung verglichen.

[Vergleichen mit Projektdatenbank]	[Compare with ENI Project]	Falls das Projekt in einer ENI-Datenbank verwaltet wird, können Sie die lokal geöffnete Version mit der aktuellen Datenbankversion vergleichen.
[Leerzeichen ignorieren]	[Ignore whitespaces]	Wenn die Option [Leerzeichen ignorieren] aktiviert ist, werden keine Unterschiede gemeldet, die in einer unterschiedlichen Anzahl von Leerzeichen bestehen.
[Kommentare ignorieren]	[Ignore comments]	Wenn die Option [Kommentare ignorieren] aktiviert ist, werden keine Unterschiede gemeldet, die Kommentare betreffen.
[Eigenschaften ignorieren]	[Ignore properties]	Wenn die Option [Eigenschaften ignorieren] aktiviert ist, werden keine Unterschiede gemeldet, die die Objekt-Eigenschaften betreffen.
[Änderungen gegenüberstellen]	[Compare differences]	Wenn die Option aktiviert ist: Wenn eine Einheit innerhalb eines Bausteins nicht gelöscht oder neu hinzugefügt, sondern nur verändert wurde, wird im zweigeteilten Fenster des Vergleichsmodus die Version des Vergleichsprojekts der des aktuellen Projekts gegenübergestellt und rot markiert (siehe Bild).  Wenn die Option deaktiviert ist: Die betreffende Einheit wird im Vergleichsprojekt als "nicht mehr vorhanden" und im aktuellen Projekt als "neu eingefügt" dargestellt (siehe Bild). Die Einheiten werden also nicht direkt gegenübergestellt. 

Wenn Sie den Dialog [Projektvergleich] mit [OK] schließen, wird der Vergleich gemäß den aktivierten Optionen durchgeführt.

Weiterführende Informationen finden Sie auf Seite 137 (→ Abschnitt. Darstellung des Vergleichsergebnisses).

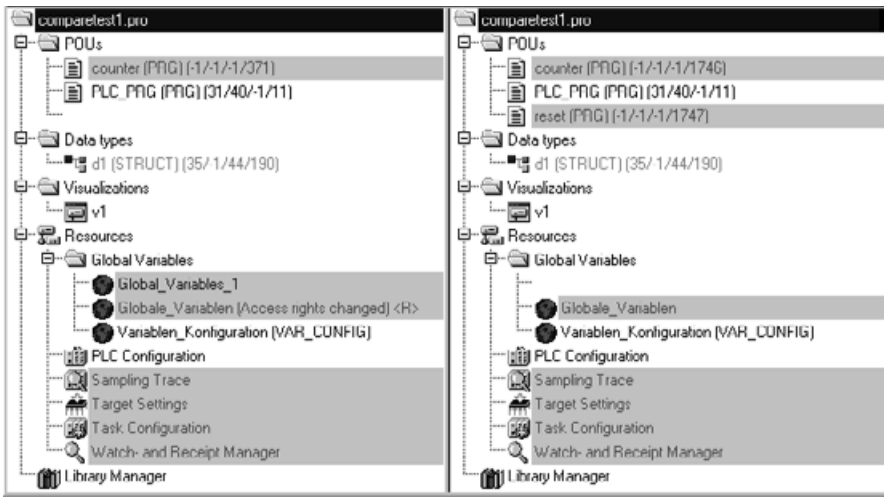
Darstellung des Vergleichsergebnisses

Das Vergleichsergebnis wird zunächst im Strukturbaum des Projekts (Projektübersicht) dargestellt, von dem aus Sie einzelne Bausteine öffnen können, um deren inhaltliche Veränderungen zu betrachten.

Projektübersicht im Vergleichsmodus

Nach dem Projektvergleich wird das Fenster [Projektvergleich] ([Project comparison]) geöffnet. In der Titelleiste stehen zusätzlich noch die Verzeichnispfade der beiden Projekte.

Die linke Fensterhälfte zeigt das aktuelle Projekt, die rechte das Vergleichsprojekt. An oberster Stelle steht jeweils der Projektname. Ansonsten entspricht die Darstellung der Struktur im **Object Organizer**:



20740AEN

Bausteine, die Unterschiede aufweisen, werden mit einer Schattierung hinterlegt und durch die Textfarbe bzw. einen Textzusatz gekennzeichnet:

Markierung	Bedeutung
Rot	Einheit wurde modifiziert; wird in beiden Fensterhälften rot dargestellt.
Blau	Einheit ist nur im Vergleichsprojekt vorhanden; an der gegenüberliegenden Stelle in Strukturbaum des aktuellen Projekts ist eine Lücke.
Grün	Einheit ist nur im aktuellen Projekt vorhanden; an der gegenüberliegenden Stelle in Strukturbaum des Vergleichsprojekts ist eine Lücke.
Schwarz	Einheit ohne Unterschiede.
"(Eigenschaften geändert)" "(Properties changed)"	Dieser Text erscheint im Strukturbaum des aktuellen Projekts hinter einem Bausteinnamen, wenn Unterschiede in den Bausteineigenschaften gefunden wurden.
"(Zugriffsrechte geändert)" "(Access rights changed)"	Dieser Text erscheint im Strukturbaum des aktuellen Projekts hinter einem Bausteinnamen, wenn Unterschiede in den Zugriffsrechten gefunden wurden.

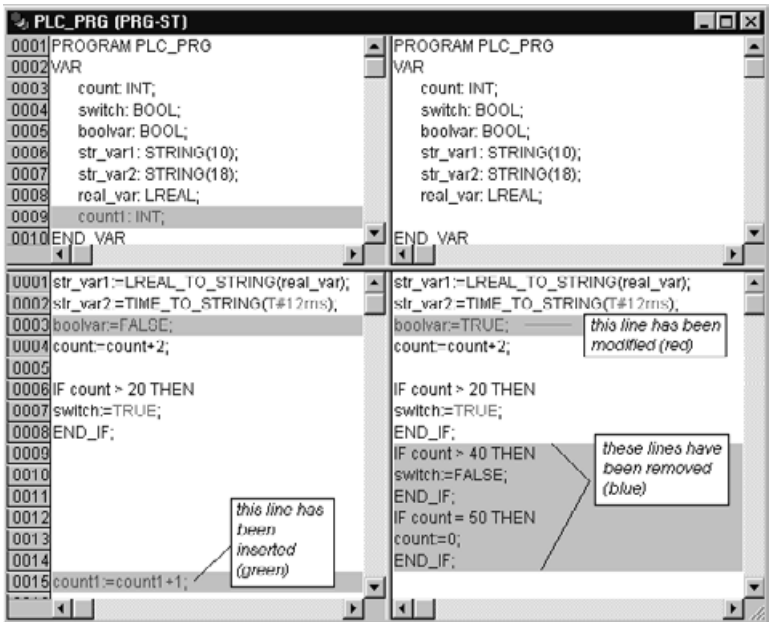
Bausteininhalt im Vergleichsmodus

Mit einem Doppelklick auf eine Zeile in der Projektübersicht öffnen Sie den dazugehörigen Baustein.

Wenn es sich um einen modifizierten (roten) Texteditor- oder Grafikeditor-Baustein handelt, wird der Baustein in einem zweigeteilten Fenster geöffnet.

Links steht der Inhalt des Bausteins aus dem aktuellen Projekt, rechts der des Bausteins aus dem Vergleichsprojekt. Für die unterschiedlichen Einheiten werden die oben be-

schriebenen farblichen Kennzeichnungen angewendet.



20741AEN

Weiterführende Informationen finden Sie auf Seite 138 (→ Abschnitt. Im Vergleichsmodus arbeiten).

Im Vergleichsmodus arbeiten

Wenn Sie im Fenster [Projektvergleich] ([Project comparison]) auf einen Eintrag mit unterschiedlichen Versionen im aktuellen Projekt und im Vergleichsprojekt klicken, stehen Ihnen im Menü [Extras] ([Extras]) bzw. im Kontextmenü (Klick mit rechter Maustaste) folgende Menübefehle zur Verfügung:

Menübefehl		Tastaturbefehl	Wirkung
[Nächster Unterschied]	[Next difference]	<F7>	Nächste Zeile mit Unterschieden wird markiert.
[Vorheriger Unterschied]	[Previous difference]	<Umschalt>+<F7>	Vorhergehende Zeile mit Unterschieden wird markiert.
[Änderung übernehmen]	[Accept change]	Leertaste	Für alle zusammenhängenden Einheiten (z. B. aufeinanderfolgende Zeilen), die die gleiche Änderungsmarkierung haben, wird die Version des Vergleichsprojekts ins aktuelle Projekt übernommen. Die betreffenden Einheiten erscheinen daraufhin in der entsprechenden Farbe in der linken Fensterhälfte. Wenn es sich um eine ursprünglich rot markierte Einheit handelt, wird die Übernahme durch gelbe Schrift im aktuellen Projekt kenntlich gemacht.
[Einzelne Änderung übernehmen]	[Accept individual changes]	<Strg>+Leertaste	Für die markierte Einheit wird die Version des Vergleichsprojekts in das aktuelle Projekt übernommen. Die Einheit erscheint daraufhin in der entsprechenden Farbe in der linken Fensterhälfte. Wenn es sich um eine ursprünglich rot markierte Einheit handelt, wird die Übernahme durch gelbe Schrift im aktuellen Projekt kenntlich gemacht.
[Eigenschaften übernehmen]	[Accept properties]	–	Für den markierten Baustein werden die Bausteineigenschaften aus dem Vergleichsprojekt in das aktuelle Projekt übernommen.
[Zugriffsrechte übernehmen	[Accept access rights]	–	Für den markierten Baustein werden die Zugriffsrechte aus dem Vergleichsprojekt in das aktuelle Projekt übernommen.

4.3.21 [Projekt] / [Kopieren] ([Project] / [Merge])

Mit dem Menübefehl [Projekt] / [Kopieren] können Sie Objekte (Bausteine, Datentypen und Ressourcen) sowie Verknüpfungen zu Bibliotheken aus anderen Projekten in das aktuelle Projekt kopieren.

Nach dem Ausführen des Menübefehls wird der Dialog [Projekt kopieren] ([Copy Project]) geöffnet. Markieren Sie im Dateiverzeichnis die Projektdatei, aus der die Objekte kopiert werden sollen.

Bestätigen Sie mit [Öffnen] ([Open]). Der Dialog [Objekt kopieren] ([Merge Project]) wird geöffnet.

In dem Dialog [Objekt kopieren] legen Sie fest, welche Objekte kopiert werden sollen.

Wenn Sie auf ein Ordnersymbol klicken, wird der gesamte untergeordnete Bereich markiert.

Um die Auswahl auf Teilbereiche oder einzelne Objekte zu beschränken, verwenden Sie die Tasten <Umschalt> und <Strg> plus Mausklick.

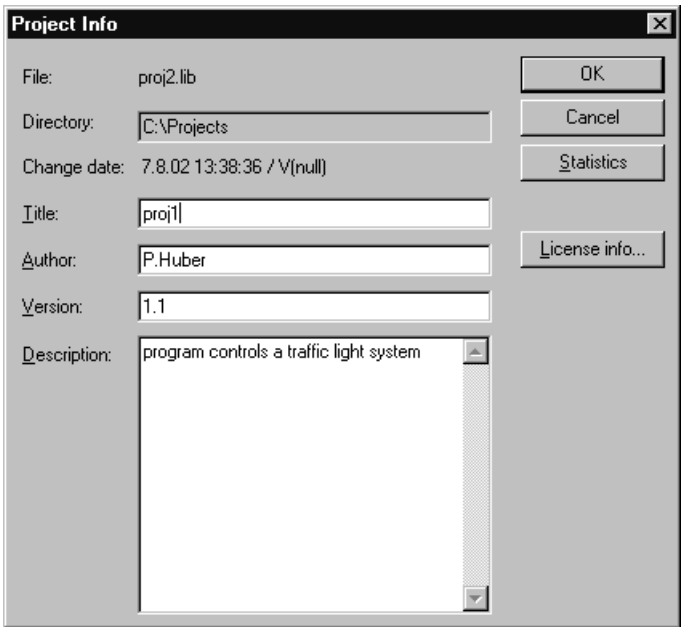
Nachdem Sie alle zu importierenden Objekte markiert haben, klicken Sie auf [OK].

Wenn ein gleichnamiges Objekt im Projekt bereits vorhanden ist, erscheint die Meldung: "Das Objekt existiert bereits. Wollen Sie es ersetzen?"

Wenn Sie auf [Ja] ([Yes]) klicken, wird das Objekt im Projekt durch das Objekt aus der Importdatei ersetzt. Wenn Sie auf [Nein] ([No]) klicken, erhält der Name des neuen Objekts als Ergänzung einen Unterstrich und eine Zählnummer (z. B. _1). Mit [Ja, alle] bzw. [Nein, alle] ([Yes, all] bzw. [No, all]) wird dies für alle Objekte bzw. kein Objekt durchgeführt.

4.3.22 [Projekt] / [Projektinformation] ([Project] / [Project Info])

Unter diesem Menüpunkt können Sie Informationen zum Projekt abspeichern. Nach Ausführen des Menübefehls [Projekt] / [Projektinformation] wird der Dialog [Projektinformation] ([Project Info]) geöffnet.



20742AEN

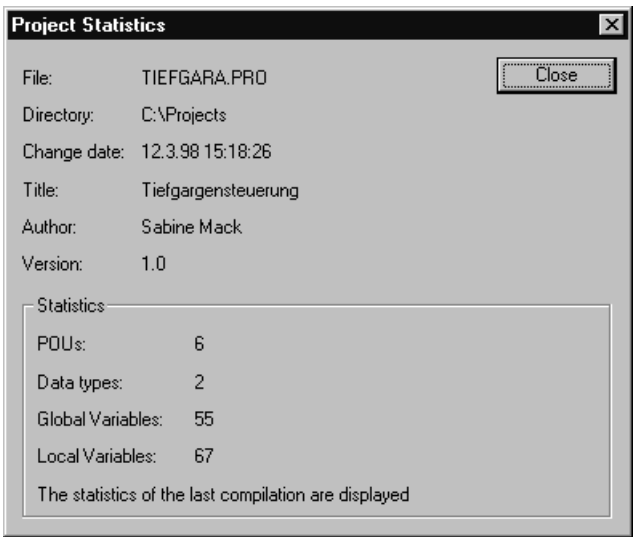
Folgende Projektinformationen werden automatisch angelegt und können nicht geändert werden:

- [Dateiname] ([File])
- [Verzeichnispfad] ([Directory])
- [Geändert am] ([Change date]): Zeitpunkt der letzten Änderung

Darüber hinaus können Sie noch folgende eigene Angaben hinzufügen:

- [Bezeichnung] ([Title])
  Hinweis: Sofern vom Zielsystem unterstützt, wird die hier eingetragene Bezeichnung automatisch als Dateiname vorgeschlagen, sobald das Projekt über den Menübefehl [Datei] / [Öffnen] / [Projekt aus der Steuerung öffnen] ([File] / [Open] / [Open project from PLC]) wieder in den PLC-Editor geladen wird. In diesem Fall wird der Dialog [Speichern] ([Save]) geöffnet.
- [Autor] ([Author])
- [Version] ([Version])
- [Beschreibung] ([Description])

Nach dem Klicken auf die Schaltfläche [Statistik] ([Statistics]) wird der Dialog [Projektstatistik] ([Project Statistics]) geöffnet. Dieser enthält die Angaben aus dem Dialog [Projektinformation] ([Project Info]) sowie die Anzahl der Bausteine, der Datentypen, der lokalen und der globalen Variablen. Die Werte beziehen sich auf den letzten Übersetzungsvorgang.



20743AEN

Wenn Sie in den Projektoptionen, Kategorie [Laden & Speichern] ([Load & Save]) die Option [Projektinformation verlangen] ([Ask for project info]) aktivieren, wird beim Abspeichern eines neuen Projekts oder beim Abspeichern eines Projekts unter einem neuen Namen automatisch der Dialog [Projektinformation] aufgerufen.

4.3.23 [Projekt] / [Global Suchen] ([Project] / [Global Search])

Mit dem Menübefehl [Projekt] / [Global Suchen] können Sie in folgenden Elementen nach Text suchen:

- Bausteinen
- Datentypen
- Objekten der globalen Variablen
- Steuerungskonfiguration
- Taskkonfiguration
- Deklarationsteilen der Bibliotheken

Nach dem Ausführen des Menübefehls wird der Dialog [Globales Suchen in] ([Global Search in]) geöffnet, in dem Sie die Objekte auswählen können, die durchsucht werden sollen.

Wenn Sie auf ein Ordnersymbol klicken, wird der gesamte untergeordnete Bereich markiert.

Um die Auswahl auf Teilbereiche oder einzelne Objekte zu beschränken, benutzen Sie die Tasten <Umschalt> und <Strg> plus Mausklick.

Beenden Sie den Dialog mit [OK].

Der Dialog [Global suchen] ([Global search]) wird geöffnet.

20744AEN

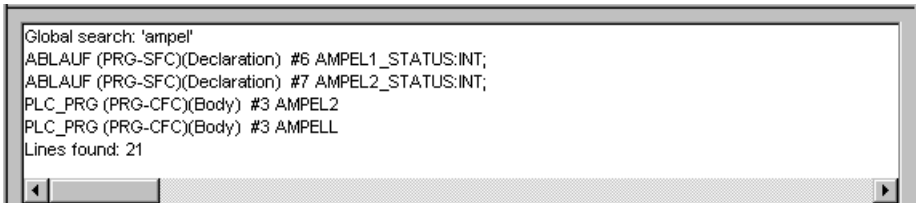
Im Eingabefeld [Suchen nach] ([Search for]) geben Sie den zu suchenden Text ein oder wählen Sie aus dem Aufklappmenü einen der vorhergehenden Suchtexte aus.

Um den Suchvorgang zu starten, drücken Sie auf die Schaltfläche [Weitersuchen] ([Find next]). An einer Fundstelle wird der Suchvorgang unterbrochen, das betreffende Objekt im Editor geöffnet und die Fundstelle markiert. Mit [Weitersuchen] können Sie den Suchvorgang fortsetzen.

Wenn Sie auf die Schaltfläche [Meldungsfenster] ([Message window]) klicken, werden alle Verwendungsstellen des gesuchten Texts in den ausgewählten Objekten im Meldungsfenster zeilenweise und in tabellarischer Form aufgelistet. Abschließend wird die Anzahl der gefundenen Stellen angegeben.

Für jede Fundstelle wird Folgendes ausgegeben:

- Objektname
- Fundstelle im Deklarationsteil (`Decl`) oder im Implementationsteil (`Impl`) eines Bausteins
- Zeilen- bzw. Netzwerknummer
- komplette Zeile bei Texteditoren
- komplette Texteinheit bei grafischen Editoren



20757AEN

Wenn Sie im Meldungsfenster auf eine Zeile doppelklicken, wird das Objekt im Editor geöffnet und der Suchtext markiert. Mit den Tasten <F4> bzw. <Umschalt>+<F4> können Sie zur vorhergehenden bzw. nachfolgenden Meldung wechseln.

Die Suchfunktion erreichen Sie auch über die Schaltfäche [ ] in der Funktionsleiste. Die Suche bezieht sich dann automatisch auf alle durchsuchbaren Projektteile.

4.3.24 [Projekt] / [Global Ersetzen] ([Project] / [Global replace])

Mit dem Menübefehl [Projekt] / [Global Ersetzen] können Sie Text in Projektteilen suchen und ersetzen. Der Menübefehl funktioniert genauso wie der Menübefehl [Projekt] / [Global Suchen] ([Project] / [Global Search]).

Ausnahme: Bibliotheken werden nicht zur Auswahl angeboten und eine Ausgabe in das Meldungsfenster ist nicht möglich.

4.3.25 [Projekt] / [Überprüfen] ([Project] / [Check])

Der Menübefehl [Projekt] / [Überprüfen] enthält vier Untermenüs:

- [Unbenutzte Variablen] ([Unused Variables])
- [Überlappende Speicherbereiche] ([Overlapping Memory Areas])
- [Konkurrierender Zugriff] ([Concurrent Access])
- [Mehrfaches Speichern auf Output] ([Multiple Write Access on Output])

Die Menübefehle dienen der Überprüfung der semantischen Korrektheit des Projekts.

Die Ergebnisse werden im Meldungsfenster ausgegeben.

Jede dieser Funktionen prüft den Stand des letzten Übersetzungslaufs. Wenn das Projekt zwischenzeitlich geändert wurde, wird eine entsprechende Warnung im Meldungsfenster ausgegeben. Um sinnvolle Prüfresultate zu erhalten, sollten Sie das Projekt vor der Prüfung neu übersetzen.

Hinweis:

Unter [Projekt] / [Optionen] / [Übersetzungsoptionen] ([Project] / [Options] / [Build]) können Sie im Bereich [Automatisch prüfen] ([Check automatically]) einstellen, dass die Prüfungen automatisch bei jedem Übersetzungslauf durchgeführt werden.

Unbenutzte Variablen

Der Menübefehl [Unbenutzte Variablen] ([Unused Variables]) des Menüs [Projekt] / [Überprüfen] ([Project] / [Check]) sucht nach Variablen, die deklariert sind, aber im Programm nicht verwendet werden. Sie werden mit Bausteinname und -zeile ausgegeben, z. B.: `PLC_PRG (4) – var1`. Variablen in Bibliotheken werden nicht berücksichtigt.

Die Ergebnisse werden im Meldungsfenster ausgegeben.

Überlappende Speicherbereiche

Der Menübefehl [Überlappende Speicherbereiche] ([Overlapping Memory Areas]) des Menüs [Projekt] / [Überprüfen] ([Project] / [Check]) prüft, ob bei der Zuweisung von Variablen zu Speicherbereichen mittels AT-Deklaration Überschneidungen entstehen.

Beispielsweise entsteht durch die Variablenzuweisungen `var1 AT %QB21: INT` und `var2 AT %QD5: DWORD` eine Überschneidung, da sie das Byte 21 gemeinsam belegen.

Sie erhalten dann folgende Meldung:

"%QB21 wird durch folgende Variablen referenziert:
PLC_PRG (3): var1 AT %QB21
PLC_PRG (7): var2 AT %QD5"

Die Ergebnisse werden im Meldungsfenster ausgegeben.

Konkurrierender Zugriff

Der Menübefehl [Konkurrierender Zugriff] ([Concurrent Access]) des Menüs [Projekt] / [Überprüfen] ([Project] / [Check]) sucht nach Speicherbereichen von IEC-Adressen, die in mehr als einer Task referenziert werden. Zwischen lesendem oder schreibendem Zugriff wird dabei nicht unterschieden.

Die Meldung kann beispielsweise so lauten:

"%MB28 wird in folgenden Tasks referenziert:
Task1 – PLC_PRG (6): %MB28 [Nur-Lese-Zugriff]
Task2 – POU1.ACTION (1) %MB28 [Schreibzugriff]"

Die Ergebnisse werden im Meldungsfenster ausgegeben.

Mehrfaches Speichern auf Output

Der Menübefehl [Mehrfaches Speichern auf Output] ([Multiple Write Access on Output]) des Menüs [Projekt] / [Überprüfen] ([Project] / [Check]) sucht nach Speicherbereichen, auf die in einem Projekt an mehr als einer Stelle schreibend zugegriffen wird.

Die Meldung kann beispielsweise so lauten:

"%QB24 wird an folgenden Stellen beschrieben:
PLC_PRG (3): %QB24
PLC_PRG.POU1 (8): %QB24"

Die Ergebnisse werden im Meldungsfenster ausgegeben.

4.3.26 Arbeitsgruppen

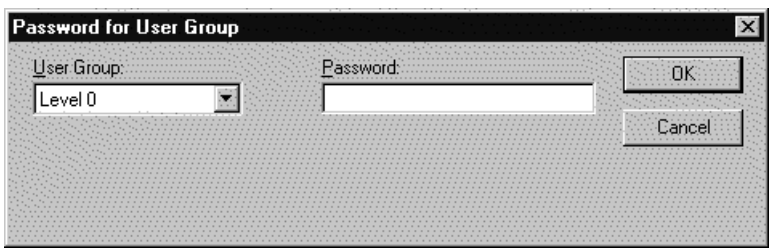
Im PLC-Editor können bis zu acht Arbeitsgruppen eingerichtet werden mit unterschiedlichen Zugriffsrechten auf Bausteine, Datentypen und Ressourcen. Die Zugriffsrechte könne für einzelne oder alle Objekte festgelegt werden. Das Projekt wird als Mitglied einer bestimmten Arbeitsgruppe geöffnet. Als solches Mitglied muss man sich mit einem Passwort autorisieren.

Die Arbeitsgruppen sind von 0 bis 7 durchnumeriert, wobei nur die Arbeitsgruppe 0 Administratorrechte besitzt. Nur Mitglieder der Arbeitsgruppe 0 dürfen Passwörter und Zugriffsrechte für alle Arbeitsgruppen und Objekte festlegen.

Wenn ein neues Projekt angelegt wird, sind zunächst alle Passwörter leer. Solange kein Passwort für die Arbeitsgruppe 0 festgelegt wurde, öffnen Sie das Projekt automatisch als Mitglied der Arbeitsgruppe 0.

Wenn für die Arbeitsgruppe 0 ein Passwort festgelegt ist, wird beim Öffnen des Projekts für alle Gruppen die Eingabe eines Passworts verlangt.

Dazu erscheint der Dialog [Passwort für Arbeitsgruppe] ([Password for User Group]):



20290AEN

In der Liste [Arbeitsgruppe] ([User group]) wählen Sie die Arbeitsgruppe aus und geben im Eingabefeld rechts das Passwort ein. Bestätigen Sie mit [OK].

Wenn das Passwort nicht mit dem gespeicherten Passwort übereinstimmt, wird folgende Meldung angezeigt: "Das Kennwort ist nicht korrekt."

Wenn das Passwort korrekt ist, wird das Projekt geöffnet.

Achtung:

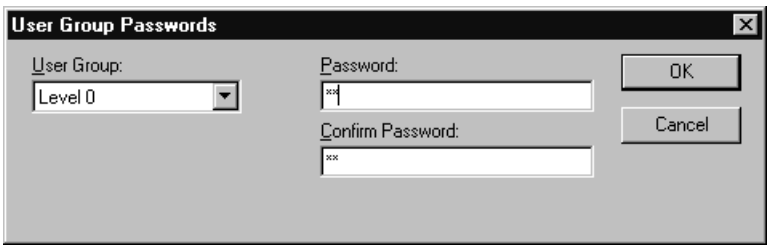
Wenn nicht für alle Arbeitsgruppen Passwörter vergeben werden, kann ein Benutzer ein Projekt über eine Arbeitsgruppe, für die kein Passwort vergeben wurde, öffnen und hat dann Administratorrechte!

Mit dem Menübefehl [Projekt] / [Passwörter für Arbeitsgruppe] ([Project] / [User Group Passwords]) können Sie die Passwörter vergeben und mit dem Menübefehl [Projekt] / [Objekt] / [Eigenschaften] ([Project] / [Objects] / [Properties]) die Zugriffsrechte für einzelne oder alle Objekte festlegen.

Passwörter für Arbeitsgruppen

Mit dem Menübefehl [Projekt] / [Passwörter für Arbeitsgruppe] ([Project] / [User Group Passwords]) vergeben Sie Passwörter für die Arbeitsgruppen. Dieser Menübefehl kann nur von Mitgliedern der Arbeitsgruppe 0 ausgeführt werden.

Nach dem Ausführen des Menübefehls [Projekt] / [Passwörter für Arbeitsgruppe] wird der Dialog [Passwort für Arbeitsgruppe] ([User group passwords]) geöffnet.



20291AEN

1. Wählen Sie in der Liste [Arbeitsgruppe] ([User group]) eine Arbeitsgruppe aus.
2. Geben Sie im Eingabefeld [Passwort] ([Password]) ein Passwort ein. Für jedes eingegebene Zeichen erscheint im Eingabefeld ein Sternchen "*".
3. Wiederholen Sie das Passwort im Eingabefeld [Passwort bestätigen] ([Confirm Password]) ein.
4. Bestätigen Sie den Dialog mit [OK].

Wenn die Passwörter in den beiden Eingabefeldern nicht übereinstimmen, erscheint die Meldung: "Das Kennwort und seine Bestätigung stimmen nicht überein."

Korrigieren Sie Ihre Eingaben und schließen Sie den Dialog mit [OK].

Wiederholen Sie die Arbeitsschritte für die nächste Arbeitsgruppe.

Achtung:

Wenn nicht für alle Arbeitsgruppen Passwörter vergeben werden, kann ein Benutzer ein Projekt über eine Arbeitsgruppe, für die kein Passwort vergeben wurde, öffnen und hat dann Administratorrechte!

Mit dem Menübefehl [Projekt] / [Objekt] / [Eigenschaften] ([Project] / [Objects] / [Properties]) können Sie die Zugriffsrechte für einzelne oder alle Objekte festlegen.

4.3.27 [Projekt] / [Projektdatenbank] ([Project] / [Project database])

Übersicht

Der Menüpunkt [Projekt] / [Projektdatenbank] steht zur Verfügung, wenn in den Projektoptionen, Kategorie [Projektdatenbank] ([Database-connection]) die Option [Projektdatenbank (ENI) verwenden] ([Use source control]) aktiviert ist.

Der Menüpunkt enthält ein Untermenü mit Befehlen zur Verwaltung des Objekts bzw. Projekts in der aktuell über die ENI-Schnittstelle verknüpften Datenbank.

Das Untermenü lässt sich in drei Bereiche einteilen:

- Login
- Menübefehle, die nur das markierte Objekt betreffen
- Menübefehle, die alle Objekte des Projekts betreffen

Login

[Login] ([Login]):
Anmelden des Benutzers beim ENI-Server

Menübefehle, die nur das markierte Objekt betreffen

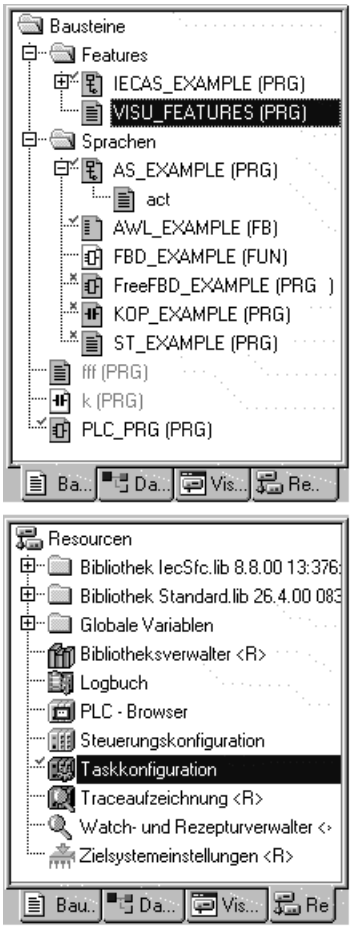
Die Menübefehle können nur ausgeführt werden, wenn Sie in die Projektdatenbank eingeloggt sind.

- [Festlegen] ([Define]), Seite 147
- [Abrufen] ([Get latest version]), Seite 148
- [Auschecken] ([Check Out]), Seite 148
- [Einchecken] ([Check In]), Seite 148
- [Auschecken rückgängig] ([Undo Check Out]), Seite 148
- [Unterschied anzeigen] ([Show differences]), Seite 148
- [Versionsgeschichte anzeigen] ([Show Version History]), Seite 149

Menübefehle, die alle Objekte des Projekts betreffen

- [Mehrfach festlegen] ([Multiple define]), Seite 150
- [Alles abrufen] ([Get All Latest Versions]), Seite 150
- [Mehrfach auschecken] ([Multiple Check Out]), Seite 150
- [Mehrfach einchecken] ([Multiple Check In]), Seite 151
- [Mehrfach auschecken rückgängig] ([Undo Multiple Check Out]), Seite 151
- [Projekt Versionsgeschichte] ([Project Version History]), Seite 151
- [Version labeln] ([Label Version]), Seite 153
- [Gemeinsame Objekte einfügen] ([Add Shared Objects]), Seite 153
- [Status auffrischen] ([Refresh Status]), Seite 154

Darstellung des Objektstatus im Object Organizer bezüglich der Verwaltung in der Projektdatenbank



Grau schattiertes Icon:
Objekt wird in der Datenbank verwaltet.

Grüner Haken vor Objektnamen:
Objekt wurde vom aktuell geöffneten Projekt aus ausgecheckt.

Rotes Kreuz vor Objektnamen:
Objekt ist momentan von einem anderen Benutzer ausgecheckt.

<R> hinter Objektnamen:
Auf das Objekt kann nur lesend zugegriffen werden.

Hinweis
Einige Objekte sind grundsätzlich mit einem <R> versehen, solange sie nicht ausgecheckt sind:
- Taskkonfiguration
- Tracekonfiguration
- Steuerungskonfiguration
- Zielsystemeinstellungen
- Watch- und Rezepturverwalter

Das bedeutet, dass keine automatische Abfrage "Objekt auschecken?" erscheint, wenn mit dem Editieren des Objekts begonnen wird.
Die Kennzeichnung mit <R>> bedeutet jedoch nicht automatisch, dass kein Schreibzugriff möglich ist. Wenn kein Schreibzugriff möglich ist, ist der Menübefehl [Auschecken] nicht ausführbar.

Projektdatenbank festlegen

Menübefehl: [Projekt] / [Projektdatenbank] / [Festlegen] ([Project] / [Project database] / [Define])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Festlegen] legen Sie fest, ob das im **Object Organizer** markierte Objekt in der Datenbank oder nur lokal im Projekt verwaltet werden soll.

Nach dem Ausführen des Menübefehls wird der Dialog [Objekteigenschaften] ([Object properties]) geöffnet.

In dem Dialog können Sie das Objekt mit einer der folgenden Datenbanken verbinden:

- [Projekt] ([Project])
- [Gemeinsame Objekte] ([Shared Objects])
- [Lokal] ([Local])

Die Symbole aller Objekte, die in der Datenbank verwaltet werden, erscheinen im Object Organizer grau schattiert. Gemeinsame Objekte werden in türkisfarbener Schrift dargestellt.

Objekt abrufen

Menübefehl: [Projekt] / [Projektdatenbank] / [Abrufen] ([Project] / [Project database] / [Get Latest Version])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Abrufen] wird die aktuelle Version des im Object Organizer markierten Objekts aus der Datenbank abgerufen und damit die lokale Version des Objekts ersetzt. Im Gegensatz zum Auschecken wird das Objekt in der Datenbank nicht für die Bearbeitung durch andere Benutzer gesperrt.

Objekt auschecken

Menübefehl: [Projekt] / [Projektdatenbank] / [Auschecken] ([Project] / [Project database] / [Check out])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Auschecken] wird das im Object Organizer markierte Objekt aus der Datenbank ausgecheckt und dadurch für die Bearbeitung durch andere Benutzer gesperrt.

Nach dem Ausführen des Menübefehls [Projekt] / [Projektdatenbank] / [Auschecken] wird der Dialog [Objekt auschecken] ([Check out object]) geöffnet. Sie können in dem Dialog einen Kommentar eingeben, der in der Versionsgeschichte des Objekts in der Datenbank zusammen mit dem Auscheckvorgang gespeichert wird. Zeilenumbrüche fügen Sie mit dem Tastaturbefehl <Strg>+<Eingabe> ein. Wenn sich die Version des Objekts in der Datenbank von der im lokalen Projekt unterscheidet, wird eine Meldung ausgegeben und Sie können entscheiden, ob Sie dennoch auschecken möchten.

Bestätigen Sie den Dialog mit [OK]. Das Objekt wird im Object Organizer mit einem grünen Haken vor dem Objektnamen markiert. Dieser Haken bedeutet, dass das Objekt ausgecheckt wurde und bearbeitet werden kann. Bei anderen Benutzern wird das Objekt mit einem roten Kreuz markiert und für die Bearbeitung gesperrt.

Objekt einchecken

Menübefehl: [Projekt] / [Projektdatenbank] / [Einchecken] ([Project] / [Project database] / [Check In])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Einchecken] wird das im Object Organizer markierte Objekt in die Datenbank eingecheckt. Dabei wird in der Datenbank eine neue Version des Objekts angelegt. Die bisherigen Versionen bleiben erhalten.

Nach dem Ausführen des Menübefehls [Projekt] / [Projektdatenbank] / [Einchecken] wird der Dialog [Objekt einchecken] ([Check in object]) geöffnet. Sie können in dem Dialog einen Kommentar eingeben, der in der Versionsgeschichte des Objekts in der Datenbank zusammen mit dem Auscheckvorgang gespeichert wird.

Bestätigen Sie den Dialogs mit [OK]. Daraufhin verschwindet im Object Organizer der grüne Haken vor dem Objektnamen.

Objekt – Auschecken rückgängig

Menübefehl: [Projekt] / [Projektdatenbank] / [Auschecken rückgängig] ([Project] / [Project database] / [Undo Check Out])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Auschecken rückgängig] wird das Auschecken des im Object Organizer markierten Objekts rückgängig gemacht und die lokal in diesem Objekt vorgenommenen Änderungen entfernt.

Das Objekt bleibt in unveränderter Version in der Datenbank und ist für andere Bearbeiter zur Bearbeitung wieder freigegeben. Im Object Organizer verschwindet der grüne Haken vor dem Objektnamen.

Objekt – Unterschiede anzeigen

Menübefehl: [Projekt] / [Projektdatenbank] / [Unterschiede anzeigen] ([Project] / [Project database] / [Show Differences])

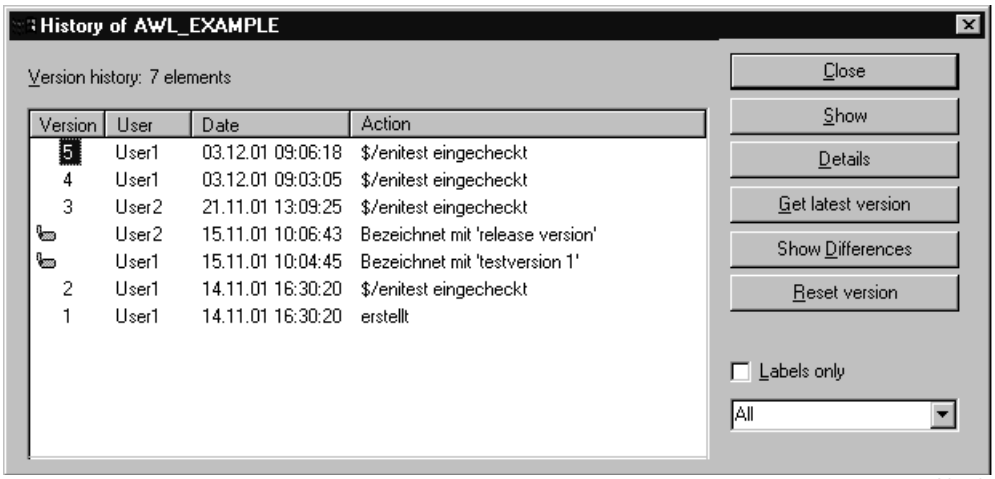
Nach dem Ausführen des Menübefehls [Projekt] / [Projektdatenbank] / [Unterschiede anzeigen] wird das gerade bearbeitete Objekt in einem zweigeteilten Fenster dargestellt. In dem Fenster ist die lokale, bearbeitete Version der aktuellen Version aus der Datenbank gegenüberstellt. Die Unterschiede der Versionen werden optisch wie beim

Menübefehl [Projekt] / [Vergleichen] ([Projekt] / [Compare]) dargestellt.

Objekt – Versionsgeschichte anzeigen

Menübefehl: [Projekt] / [Projektdatenbank] / [Versionsgeschichte anzeigen] ([Project] / [Project database] / [Show Version History])

Nach dem Ausführen des Menübefehls [Projekt] / [Projektdatenbank] / [Versionsgeschichte anzeigen] wird der Dialog [Versionsgeschichte von <Objektname>] ([History of <Objectname>]) geöffnet. In dem Dialog sind zusätzlich zum aktuell bearbeiteten Objekt alle Versionen aufgelistet, die in die Datenbank eingecheckt bzw. mit einem Bezeichner (Label) versehen wurden.



20579AEN

Folgende Angaben zu den einzelnen Versionen werden dargestellt:

Eintrag		Bedeutung
[Version]	[Version]	Datenbankabhängige Nummerierung der zeitlich nacheinander eingecheckten Versionen des Objekts. Mit einem Bezeichner (Label) versehene Versionen erhalten keine Versionsnummer, sondern sind mit einem Icon gekennzeichnet.
[Benutzer]	[User]	Name des Benutzers, der die Aktion am Objekt durchgeführt hat.
[Datum]	[Date]	Datum und Uhrzeit der Aktion
[Aktion]	[Action]	Art der Aktion, die am Objekt durchgeführt wurde. Die Form des Eintrags ist datenbankabhängig, z. B.: • "erstellt": Das Objekt wurde in der Datenbank erstmals eingecheckt • "eingecheckt" • "Bezeichnet mit <label>": Diese Version des Objekts wurde mit einem Bezeichner (Label) versehen.

Die Schaltflächen und Optionen im Dialog haben folgende Bedeutung:

Schaltfläche/Option		Bedeutung
[Schließen]	[Close]	Der Dialog wird geschlossen.
[Anzeigen]	[Show]	Die in der Liste markierte Version wird im PLC-Editor in einem Fenster geöffnet. In der Titelleiste des Fensters steht: `ENI: <Name des Projekts in der Datenbank>/<Objektname>`.
[Details]	[Details]	Der Dialog [Details der Versionsgeschichte] ([Details to Version History]) wird geöffnet. Der Dialog enthält folgende Informationen: • [Datei] ([File]): Name des Projekts und des Objekts in der Datenbank • [Version] ([Version]): siehe vorherige Tabelle • [Datum] ([Date]): siehe vorherige Tabelle • [Benutzer] ([User]): siehe vorherige Tabelle • [Kommentar] ([Comment]): Kommentar, der beim Einchecken bzw. Bezeichnen (Labeln) eingegeben wurde. • Über die Schaltflächen [Nächste] bzw. [Vorherige] ([Next] bzw. [Previous]) wechseln Sie zwischen den Versionen des Objekts.
[Abrufen]	[Get latest version]	Die in der Liste markierte Version wird aus der Datenbank in den PLC-Editor geladen und ersetzt die lokale Version des Objekts.

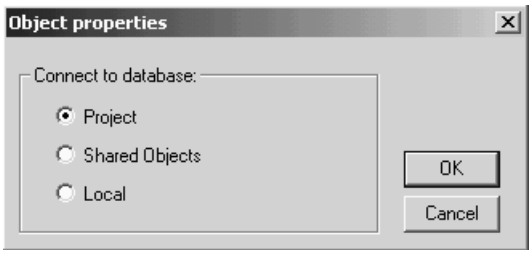
Schaltfläche/Option		Bedeutung
[Unterschiede]	[Show Differences]	Wenn in der Liste nur eine Version des Objekts markiert ist, wird diese Version mit der aktuellen Datenbankversion verglichen. Wenn zwei Versionen markiert sind, werden diese miteinander verglichen. Die Unterschiede werden wie beim Projektvergleich in einem zweigeteilten Fenster dargestellt.
[Version zurücksetzen]	[Reset version]	Die in der Liste markierte Version wird zur aktuellen Datenbankversion. Alle später eingefügten Versionen werden gelöscht!
[Nur Bezeichnungen]	[Labels only]	Wenn Sie die Option [Nur Bezeichnungen] aktivieren, werden nur die mit einem Bezeichner (Label) versehenen Versionen in der Liste angezeigt. In der darunterliegenden Auswahlliste können Sie die Anzeige auf einen bestimmten Benutzer beschränken. Wählen Sie dazu einen Benutzer aus der Liste aus. Standardmäßig ist die Einstellung [Alle] ([All]).

Projektdatenbank mehrfach festlegen

Menübefehl: [Projekt] / [Projektdatenbank] / [Mehrfach festlegen] ([Project] / [Project database] / [Multiple Define])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Mehrfach festlegen] kann für mehrere Objekte des aktuellen Projekts gleichzeitig festgelegt werden, in welcher Projektdatenbank sie verwaltet werden sollen.

Nach dem Ausführen des Menübefehls [Projekt] / [Projektdatenbank] / [Mehrfach festlegen] wird der Dialog [Objekteigenschaften] ([Object properties]) geöffnet.



20580AEN

Wählen Sie die gewünschte Projektdatenbank und schließen den Dialog mit [OK].

Daraufhin wird der Dialog [ENI-Auswahl] ([ENI-Selection]) geöffnet, in dem die Objekte des Projekts aufgelistet sind, die mit der eingestellten Projektdatenbank verbunden werden können. Die Darstellung entspricht der im Object Organizer verwendeten Baumstruktur.

Markieren Sie die gewünschten Objekte und bestätigen Sie mit [OK].

Alles abrufen

Menübefehl: [Projekt] / [Projektdatenbank] / [Alles abrufen] ([Project] / [Project database] / [Get All Latest Versions])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Alles abrufen] wird für das geöffnete Projekt die aktuelle Version aller Objekte der Projektdatenbank "Projekt" abgerufen.

Wenn in der Projektdatenbank Objekte hinzugefügt wurden, werden diese lokal eingefügt.

Wenn in der Projektdatenbank Objekte gelöscht wurden, werden diese lokal nicht gelöscht, aber automatisch der Projektdatenbank "Lokal" zugeordnet.

In der Kategorie "Ressourcen" werden nur diejenigen Objekte aus dem Projekt abgerufen, die bereits im lokalen Projekt angelegt sind. Weiterführende Informationen zur Bedeutung des Abrufens finden Sie auf Seite 148 (→ Abschnitt. Objekt abrufen).

Mehrfach auschecken

Menübefehl: [Projekt] / [Projektdatenbank] / [Mehrfach auschecken] ([Project] / [Project database] / [Multiple Check Out])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Mehrfach auschecken] können mehrere Objekte gleichzeitig ausgecheckt werden.

Nach dem Ausführen des Menübefehls wird der Dialog [ENI-Auswahl] ([ENI-Selection]) geöffnet, in dem die Objekte des Projekts aufgelistet sind. Die Darstellung entspricht der im Object Organizer verwendeten Baumstruktur.

Markieren Sie die Objekte, die ausgecheckt werden sollen, und bestätigen Sie mit [OK].

Weiterführende Informationen zur Bedeutung des Auscheckens finden Sie auf Seite 148 (→ Abschnitt. Objekt auschecken).

Mehrfach einchecken

Menübefehl: [Projekt] / [Projektdatenbank] / [Mehrfach einchecken] ([Project] / [Project database] / [Multiple Check In])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Mehrfach einchecken] können mehrere Objekte gleichzeitig eingecheckt werden.

Nach dem Ausführen des Menübefehls wird der Dialog [ENI-Auswahl] ([ENI-Selection]) geöffnet, in dem die ausgecheckten Objekte des Projekts aufgelistet sind. Die Darstellung entspricht der im **Object Organizer** verwendeten Baumstruktur.

Markieren Sie die Objekte, die eingecheckt werden sollen, und bestätigen Sie mit [OK].

Weiterführende Informationen zur Bedeutung des Eincheckens finden Sie im (→ Abschnitt. Objekt einchecken), Seite 148.

Mehrfach auschecken rückgängig

Menübefehl: [Projekt] / [Projektdatenbank] / [Mehrfach auschecken rückgängig] ([Project] / [Project database] / [Undo Multiple Check Out])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Mehrfach auschecken rückgängig] kann für mehrere Objekte gleichzeitig das Auschecken rückgängig gemacht werden.

Nach dem Ausführen des Menübefehls wird der Dialog [ENI-Auswahl] ([ENI-Selection]) geöffnet, in dem die ausgecheckten Objekte des Projekts aufgelistet sind. Die Darstellung entspricht der im Object Organizer verwendeten Baumstruktur.

Markieren Sie die Objekte, bei denen das Auschecken rückgängig gemacht werden soll, und bestätigen Sie mit [OK].

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Mehrfach auschecken rückgängig] wird das Auschecken der markierten Objekte rückgängig gemacht und die lokal in diesen Objekten vorgenommenen Änderungen entfernt.

Das Objekt bleibt in unveränderter Version in der Datenbank und ist für andere Bearbeiter zur Bearbeitung wieder freigegeben. Im Object Organizer verschwindet der grüne Haken vor dem Objektnamen.

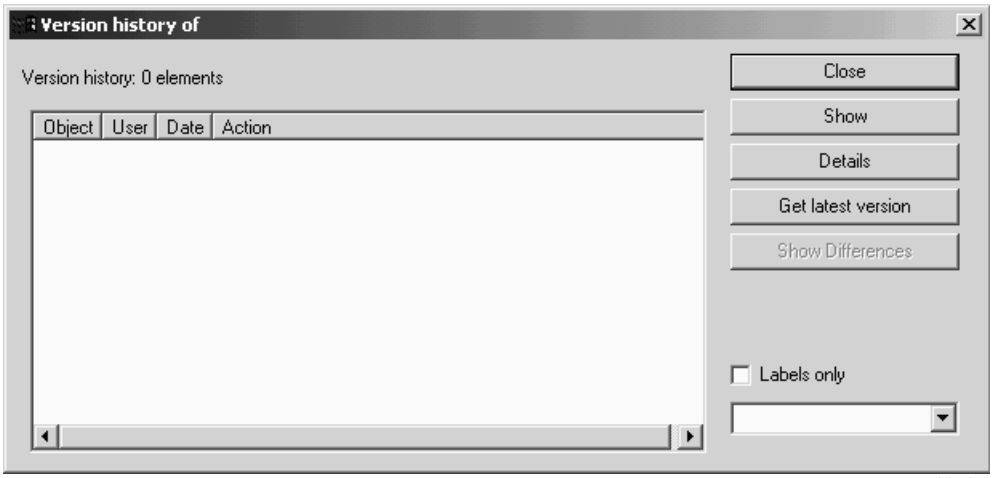
Projekt – Versionsgeschichte

Menübefehl: [Projekt] / [Projektdatenbank] / [Projekt Versionsgeschichte] ([Project] / [Project database] / [Project Version History])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Projekt Versionsgeschichte] können Sie die Versionsgeschichte des aktuellen Projekts einsehen.

Nach dem Ausführen des Menübefehls wird der Dialog [Versionsgeschichte von <Projektname in Projektdatenbank>] ([Version history of <Project name in project data base>]) geöffnet, in dem in chronologischer Folge die Aktionen (Erstellen, Einchecken, Bezeichnen) für alle zum Projekt gehörenden Objekte aufgelistet sind. Die Anzahl die-

ser Objekte wird hinter [Versionsgeschichte] ([Version history]) angegeben.



20581AEN

Folgende Angaben zur den einzelnen Objekten werden dargestellt:

Eintrag		Bedeutung
[Objekt]	[Object]	Name des Objekts
[Benutzer]	[User]	Name des Benutzers, der die Aktion am Objekt durchgeführt hat.
[Datum]	[Date]	Datum und Uhrzeit der Aktion
[Aktion]	[Action]	Art der Aktion, die am Objekt durchgeführt wurde. Die Form des Eintrags ist datenbankabhängig, z. B.: • "erstellt": Das Objekt wurde in der Datenbank erstmals eingecheckt • "eingecheckt" • "Bezeichnet mit <label>": Diese Version des Objekts wurde mit einem Bezeichner versehen .

Die Schaltflächen und Optionen im Dialog haben folgende Bedeutung:

Schaltfläche/Option		Bedeutung
[Schließen]	[Close]	Der Dialog wird geschlossen.
[Anzeigen]	[Show]	Die in der Liste markierte Version wird im PLC-Editor in einem Fenster geöffnet. In der Titelleiste des Fensters steht: `ENI: <Name des Projekts in der Datenbank>/<Objektname>`
[Details]	[Details]	Der Dialog [Details der Versionsgeschichte] ([Details to Version History]) wird geöffnet. Der Dialog enthält folgende Informationen: • [Datei] ([File]): Name des Projekts und des Objekts in der Datenbank • [Version] ([Version]): Datenbankabhängige Nummerierung der zeitlich nacheinander eingecheckten Versionen des Objekts. Mit einem Bezeichner (Label) versehene Versionen erhalten keine Versionsnummer, sondern sind mit einem Icon gekennzeichnet. • [Datum] ([Date]): siehe vorherige Tabelle • [Benutzer] ([User]): siehe vorherige Tabelle • [Kommentar] ([Comment]): Kommentar, der beim Einchecken bzw. Labeln eingegeben wurde. • Über die Schaltflächen [Nächste] bzw. [Vorherige] ([Next] bzw. [Previous]) wechseln Sie zwischen den Versionen des Objekts.
[Abrufen]	[Get latest version]	Alle Objekte aus der in der Liste markierten Projektversion werden in das lokale Projekt abgerufen. Lokale Objekte werden mit der älteren Version überschrieben. Lokale Objekte, die in dieser älteren Version noch nicht im Projekt enthalten waren, werden nicht aus der lokalen Version entfernt! Wenn eine Version mit Bezeichner (Label) abgerufen wird, die gemeinsame Objekte enthält, wird ein Dialog geöffnet. In dem Dialog können Sie festlegen, ob diese Objekte ebenfalls abgerufen werden sollen.

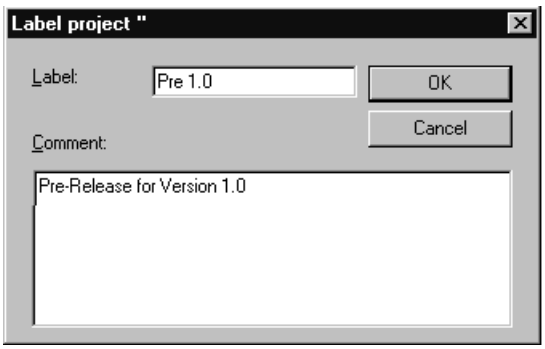
Schaltfläche/Option		Bedeutung
[Unterschiede]	[Show Differences]	Wenn in der Liste nur eine Version des Objekts markiert ist, wird diese Version mit der aktuellen Datenbankversion verglichen. Wenn zwei Versionen markiert sind, werden diese miteinander verglichen. Die Unterschiede werden wie beim Projektvergleich in einem zweigeteilten Fenster dargestellt.
[Nur Bezeichnungen]	[Labels only]	Wenn Sie die Option [Nur Bezeichnungen] aktivieren, werden nur die mit einem Bezeichner (Label) versehenen Objektversionen in der Liste angezeigt. In der darunterliegenden Auswahlliste können Sie die Anzeige auf einen bestimmten Benutzer beschränken. Wählen Sie dazu einen Benutzer aus der Liste aus. Standardmäßig ist die Einstellung [Alle] ([All]).

Projekt – Version labeln

Menübefehl: [Projekt] / [Projektdatenbank] / [Version labeln] ([Project] / [Project database] / [Label version])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Version labeln] wird der aktuelle Projektstand der Objekte unter einem Bezeichner zusammengefasst. Mit diesem Bezeichner können Sie den Projektstand später wieder abzurufen.

Nach dem Ausführen des Menübefehls wird der Dialog [Projektstand von <Projektname in der Projektdatenbank> bezeichnen] ([Label project <project name in project data base>]) geöffnet.



20582AEN

Geben Sie im Eingabefeld [Bezeichnung] ([Label]) eine Bezeichnung für den Projektstand ein. Außerdem können Sie einen Kommentar zum Projektstand schreiben.

Schließen Sie den Dialog mit [OK]. Die Bezeichnung und die Aktion des Bezeichnens ("bezeichnet mit ...") erscheinen in der Liste der Versionsgeschichte sowohl eines Objekts als auch des Projekts. Außerdem wird unter [Version] ([Version]) ein Icon angezeigt.

Gemeinsame Objekte einfügen

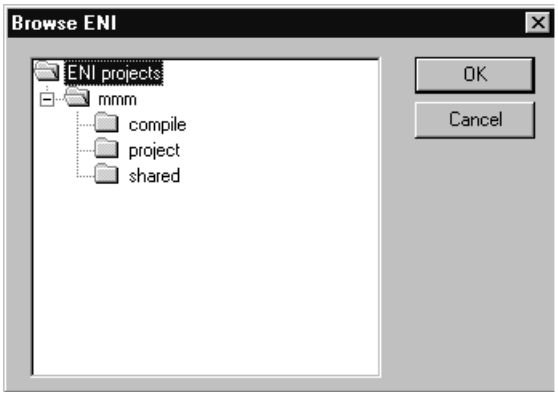
Menübefehl: [Projekt] / [Projektdatenbank] / [Gemeinsame Objekte einfügen] ([Project] / [Project database] / [Add Shared Objects])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Gemeinsame Objekte einfügen] werden zusätzliche Objekte der Projektdatenbank "Gemeinsame Objekte" in das lokal geöffnete Projekt eingebunden.

Bei Objekten der Projektdatenbank "Projekt" ist dies nicht nötig, da beim Menübefehl [Projekt] / [Projektdatenbank] / [Alles abrufen] ([Project] / [Project database] / [Get All Latest Versions]) automatisch alle aktuell vorhandenen Datenbankobjekte ins lokale Projekt geladen werden. Das gilt auch für jene Objekte, die im Projekt noch nicht angelegt sind. Bei Objekten der Projektdatenbank "Gemeinsame Objekte" werden beim Menübefehl [Projekt] / [Projektdatenbank] / [Alles abrufen] nur die bereits im Projekt eingebundenen Objekte berücksichtigt.

Nach dem Ausführen des Menübefehls [Projekt] / [Projektdatenbank] / [Gemeinsame

Objekte einfügen] wird der Dialog [ENI durchsuchen] ([Browse ENI]) geöffnet.



20583AEN

In dem Dialog sind alle Objekte aufgelistet, die in dem angegebenen Projektverzeichnis in der Projektdatenbank liegen.

Markieren Sie das gewünschte Objekt und klicken Sie auf [OK] oder doppelklicken Sie auf das Objekt. Damit wird das Objekt in das lokal geöffnete Projekt eingefügt.

Status auffrischen

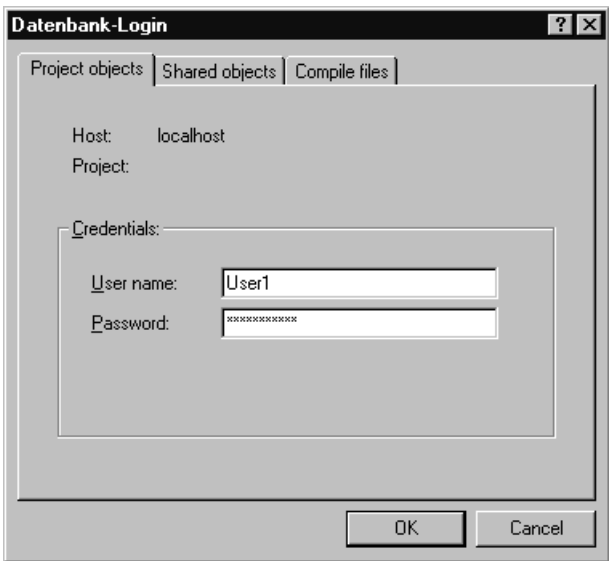
Menübefehl: [Projekt] / [Projektdatenbank] / [Status auffrischen] ([Project] / [Project database] / [Refresh Status])

Mit dem Menübefehl [Projekt] / [Projektdatenbank] / [Status auffrischen] wird die Anzeige im Object Organizer aktualisiert. Im Object Organizer ist dann der aktuelle Status der Objekte bezüglich den Projektdatenbanken dargestellt.

Login

Menübefehl: [Projekt] / [Projektdatenbank] / [Login] ([Project] / [Project database] / [Login])

Nach dem Ausführen des Menübefehls [Projekt] / [Projektdatenbank] / [Login] wird der Dialog [Datenbank-Login] ([Database Login]) geöffnet.



20584AEN

In dem Dialog müssen Sie sich beim ENI-Server für jede Projektdatenbank anmelden, um für das Projekt Verbindung zur jeweiligen Projektdatenbank zu erhalten. Die Zugangsdaten müssen also im ENI-Server (ENI-Administration, Benutzerverwaltung) und gegebenenfalls auch in der Benutzerverwaltung der Projektdatenbank bekannt sein.

Im Dialog werden folgende Angaben angezeigt:

[Projektobjekte]	[Project Objects]	Typ der Projektdatenbank
[Host]	[Host]	Rechneradresse des ENI-Servers entsprechend den Angaben in den Projektoptionen, Kategorie [Projektdatenbank] / [Projektdatenbank konfigurieren] ([Database-connection] / [Configure ENI]) im Eingabefeld [TCP/IP-Adresse] ([TCP/IP-address])
[Projekt]	[Project]	Name des Projekts in der Projektdatenbank entsprechend den Angaben in den Projektoptionen, Kategorie [Projektdatenbank] / [Projektdatenbank konfigurieren] ([Database-connection] / [Configure ENI]) im Eingabefeld [Projektname] ([Project name])

Geben Sie Benutzername und Passwort im Bereich [Legitimation] (Credentials]) ein. Wenn Sie sich als "anonymer" Benutzer einloggen wollen, lassen Sie das Eingabefeld [Benutzername] ([User name]) leer.

Um die Eingaben zu bestätigen, klicken Sie auf [OK]. Daraufhin wird der Dialog für die Projektobjekte geschlossen und der Dialog für die Projektdatenbank [Gemeinsame Objekte] ([Shared objects]) geöffnet. Geben Sie auch hier die Zugangsdaten ein und bestätigen Sie mit [OK].

Verfahren Sie abschließend genauso mit der Projektdatenbank [Übersetzungsdateien] ([Compile files]).

Der Login-Dialog wird automatisch geöffnet, sobald ein Datenbankzugriff versucht wird, bevor sich der Anwender eingeloggt hat.

Hinweis:

Wenn die hier eingegebenen Zugangsdaten zur Projektdatenbank mit dem Projekt gespeichert werden sollen, aktivieren Sie in den Projektoptionen, Kategorie [Laden & Speichern] ([Load & Save]) die Option [Zugangsdaten für Projektdatenbank speichern] ([Save ENI credentials]).

4.4 Objekte verwalten

4.4.1 Objekte

Als Objekt werden bezeichnet:

- Bausteine
- Datentypen
- Ressourcen
  - Globale Variablen
  - Variablenkonfiguration
  - Traceaufzeichnung
  - Steuerungskonfiguration
  - Taskkonfiguration
  - Watch- und Rezepturverwalter

Die zur Strukturierung des Projekts eingefügten Ordner sind zum Teil impliziert. Sämtliche Objekte eines Projekts stehen im **Object Organizer**.

Wenn Sie den Mauszeiger eine kurze Zeit über einen Baustein im Object Organizer halten, wird die Bausteinart (Programm, Funktion oder Funktionsblock) in einem Tooltip angezeigt.

Bei globalen Variablen erscheint das Schlüsselwort (`VAR_GLOBAL`, `VAR_CONFIG`).

Zusätzliche Symbole vor oder hinter den Objekteinträgen kennzeichnen den Status hinsichtlich **Online Change** und ENI-Anbindung an eine Projektdatenbank.

Mit "Drag & Drop" können Sie Objekte und Ordner innerhalb ihrer Objektart verschieben. Klicken Sie hierzu auf das Objekt und verschieben Sie es bei gedrückter linker Maustaste an den gewünschten Ort. Wenn es durch die Verschiebung zu einer Namenskollision kommt, wird das verschobene Element durch einen Unterstrich und fortlaufende Nummer hinter dem Namen eindeutig gekennzeichnet (z. B. `Objekt_1`).

4.4.2 Ordner

Um den Überblick bei größeren Projekten zu behalten, sollten Sie Bausteine, Datentypen und globale Variablen sinnvoll in Ordnern gruppieren.

Sie können die Ordner beliebig tief schachteln. Wenn vor dem geschlossenen Ordnersymbol ein Pluszeichen angezeigt wird, enthält dieser Ordner Objekte und/oder weitere Ordner.

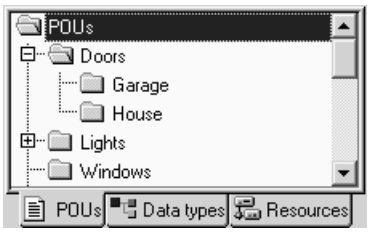
Um einen Ordner zu öffnen, doppelklicken Sie auf den Ordner. Die untergeordneten Objekte und Ordner werden angezeigt. Wenn Sie auf einen geöffneten Ordner doppelklicken, wird der Ordner wieder geschlossen. Im Kontextmenü (Klick mit rechter Maustaste) finden Sie die Befehle [Knoten expandieren] ([Expande node]) und Knoten kollabieren] ([Collapse Node]) mit der gleichen Funktionalität.

Mit "Drag & Drop" können Sie die Ordner verschieben. Klicken Sie hierzu auf den Ordner und verschieben Sie ihn bei gedrückter linker Maustaste an die gewünschte Stelle. Wenn es durch die Verschiebung zu einer Namenskollision kommt, wird der verschobene Ordner durch eine fortlaufende Nummer hinter dem Namen eindeutig gekennzeichnet (z. B. `XYOrdner 1`).

Mit dem Menübefehl [Neuer Ordner] ([New Folder]) im Kontextmenü können Sie neue Ordner einfügen.

Hinweis:

Ordner haben keinerlei Einfluss auf den Programmablauf. Sie dienen lediglich der Strukturierung Ihres Projekts.



20292AEN

Neuen Ordner einfügen

Mit dem Menübefehl [Neuer Ordner] ([New Folder]) im Kontextmenü können Sie neue Ordner einfügen. Um das Kontextmenü zu öffnen, klicken Sie im Object Organizer mit der rechten Maustaste klicken oder mit dem Tastaturbefehl <Umschalt>+<F10>.

Wenn ein Ordner markiert ist, wird der neue Ordner unter dem markierten angelegt, ansonsten auf der gleichen Ebene wie der markierte.

Wenn eine Aktion markiert ist, wird der neue Ordner auf der Ebene des Bausteins eingefügt, zu dem die Aktion gehört.

Der neue Ordner erhält zunächst die Bezeichnung `Neuer Ordner`.

Wenn bereits ein Ordner mit Namen `Neuer Ordner` auf der gleichen Hierarchieebene vorhanden ist, erhält jeder zusätzliche neue automatisch eine angehängte fortlaufende Nummer (z. B. `Neuer Ordner 1`).

Um den Ordner umzubenennen, wählen Sie im Kontextmenü den Menübefehl [Objekt umbenennen] ([Rename Object]). Der Dialog [Objekt umbenennen] ([Rename Object]) wird geöffnet, in dem Sie den neuen Namen eingeben können.

Beachten Sie folgende Namenskonventionen für Ordner:

- Ordner auf gleicher Hierarchieebene müssen unterschiedliche Namen tragen.
- Ein Ordner kann nicht den gleichen Namen erhalten wie ein Objekt, das sich auf der gleichen Hierarchieebene befindet.

4.4.3 [Knoten expandieren], [Knoten kollabieren] ([Expand Node], [Collapse Node])

Wenn ein Knoten bzw. Ordner im **Object Organizer** untergeordnete Objekte oder Ordner enthält, können Sie mit den beiden Menübefehlen [Knoten expandieren] und [Knoten kollabieren] aus dem Kontextmenü die Knoten bzw. Ordner auf- und zuklappen.

Um in das Kontextmenü zu gelangen, klicken Sie im Object Organizer mit der rechten Maustaste auf den Knoten bzw. Ordner, oder Sie markieren den Knoten bzw. Ordner und drücken Sie die Tasten <Umschalt>+<F10>.

Sie können einen Knoten bzw. Ordner außerdem durch Doppelklicken oder Drücken der <Eingabe>-Taste auf- und zuklappen.

4.4.4 [Projekt] / [Objekt] / [Löschen] ([Project] / [Object] / [Delete])

Mit dem Menübefehl [Projekt] / [Objekt] / [Löschen] (oder mit der <Entf>-Taste) löschen Sie im **Object Organizer** Objekt und Ordner. Der Menübefehl bezieht sich auf das markierte Objekt bzw. den markierten Ordner.

Bei einem Ordner werden die darunter liegenden Objekte und Ordner aus dem Object Organizer und damit aus dem Projekt entfernt.

Das Löschen können Sie mit dem Menübefehl [Bearbeiten] / [Rückgängig] ([Edit] / [Undo]) wieder rückgängig machen.

Wenn das Editorfenster des Objekts geöffnet war, wird es automatisch geschlossen.

Wenn Sie zum Löschen den Menübefehl [Bearbeiten] / [Ausschneiden] ([Edit] / [Cut]) verwenden, wird das Objekt zusätzlich in die Zwischenablage verschoben. Sie können dann das Objekt durch [Bearbeiten] / [Einfügen] ([Edit] / [Insert]) an gleicher oder anderer Stelle wieder einfügen.

4.4.5 [Projekt] / [Objekt] / [Einfügen] ([Project] / [Object] / [Add])

Mit dem Menübefehl [Projekt] / [Objekt] / [Einfügen] (oder <Einfg>-Taste) erstellen Sie ein neues Objekt. Der Objekttyp hängt von der im **Object Organizer** gewählten Registerkarte ab.

In den einzelnen Registerkarten können folgende Objekttypen eingefügt werden:

- Bausteine: Funktion, Funktionsbaustein, Programm
- Dateitypen: Dateityp
- Ressourcen: Globale Variablenliste

Beachten Sie dabei, dass für den Objekttyp gegebenenfalls eine Vorlage definiert ist.

Nach dem Auswählen des Menübefehls wird ein Dialog geöffnet, in dem Sie die benötigten Angaben zum Objekt eingeben.

Bei der Namensgebung müssen Sie folgende Einschränkungen beachten:

- Der Bausteinname darf keine Leerzeichen enthalten
- Ein Baustein darf nicht den gleichen Namen erhalten wie ein anderer Baustein bzw. wie ein Datentyp.
- Ein Datentyp darf nicht den gleichen Namen erhalten wie ein anderer Datentyp bzw. wie ein Baustein.
- Eine globale Variablenliste darf nicht den gleichen Namen erhalten wie eine andere globale Variablenliste.
- Eine Aktion darf nicht den gleichen Namen erhalten wie eine andere Aktion desselben Bausteins.

Wenn das Projekt über die ENI-Schnittstelle mit einer Projektdatenbank verknüpft ist, können Sie diese Verknüpfung so konfigurieren, dass beim Anlegen eines neuen Objekts nachgefragt wird, in welcher Projektdatenbank das Objekt verwaltet werden soll.

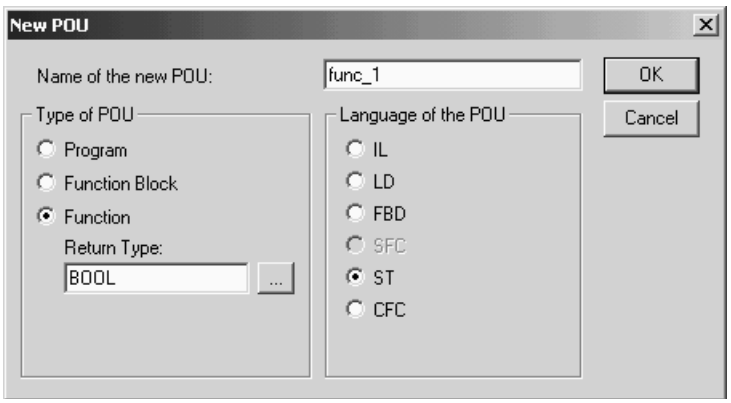
In diesem Fall wird der Dialog [Objekteigenschaften] ([Object Properties]) zur Auswahl der Projektdatenbank geöffnet.

Weitere Informationen hierzu finden Sie auf Seite 93 (→ Kap. 4.2.13, [Projektdatenbank] ([Database-connection])).

Baustein einfügen

Wechseln Sie im **Object Organizer** in die Registerkarte [Bausteine] ([POUs]) und klicken Sie an die Stelle, an der der neue Bausteine eingefügt werden soll.

Nach dem Auswählen des Menübefehls [Projekt] / [Objekt] / [Einfügen] ([Project] / [Object] / [Add]) wird der Dialog [Neuer Baustein] ([New POU]) geöffnet.



20293AEN

Geben Sie den Namen des Bausteins ein und legen Sie Typ und Programmiersprache fest.

Wenn Sie als Bausteintyp die Option [Funktion] ([Function]) aktivieren, müssen Sie im

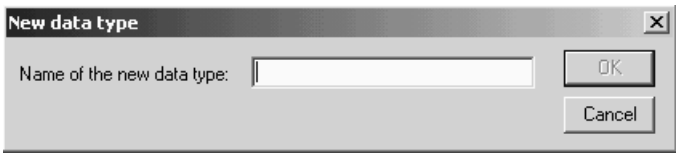
Eingabefeld [Rückgabetyp] ([Return Type]) zusätzlich den Datentyp des Rückgabewerts eingeben. Als Datentyp sind alle Standard-Datentypen und definierten Datentypen erlaubt.

Bestätigen Sie die Eingabe mit [OK]. Der neue Baustein wird im Object Organizer angelegt und im Editor geöffnet.

Datentyp einfügen

Wechseln Sie im Object Organizer in die Registerkarte [Datentypen] ([Data types]) und klicken Sie an die Stelle, an der der neue Datentyp eingefügt werden soll.

Führen Sie den Menübefehl [Projekt] / [Objekt] / [Einfügen] ([Project] / [Object] / [Add]) aus. Der Dialog [Neuer Datentyp] ([New data type]) wird geöffnet.



20294AEN

Geben Sie den Namen des Datentyps ein.

Nach dem Bestätigen der Eingabe mit [OK] wird der neue Datentyp im Object Organizer angelegt und im Editor geöffnet.

Globale Variablenliste einfügen

Informationen zur globalen Variablenliste finden Sie auf Seite 277 (→ Kap. 6.2.3, Globale Variablen).

Als Vorlage speichern

Folgende Objekttypen können als Vorlage gespeichert werden:

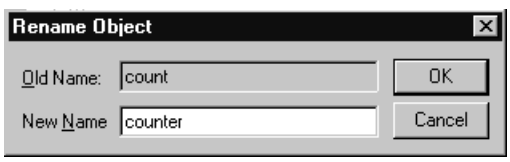
- Globale Variablen
- Dateityp
- Funktion
- Funktionsbaustein
- Programm

Markieren Sie das Objekt im Object Organizer und wählen Sie im Kontextmenü den Menübefehl [Als Vorlage speichern] ([Save as template]).

Beim Einfügen eines neuen Objekts des gleichen Typs wird der Deklarationsteil der Vorlage initial übernommen. Es wird jeweils die zuletzt angelegte Vorlage für einen Objekttyp verwendet.

4.4.6 [Projekt] / [Objekt] / [Umbenennen] ([Project] / [Object] / [Rename])

Markieren Sie das Objekt, das umbenannt werden soll. Führen Sie den Menübefehl [Projekt] / [Objekt] / [Umbenennen] aus oder betätigen Sie die <Leertaste>. Der Dialog [Objekt umbenennen] ([Rename Object]) wird geöffnet.



20295AEN

Geben Sie im Eingabefeld den neuen Namen ein. Beachten Sie hierbei die Namenskonventionen. Informationen zu den Nameskonventionen finden Sie auf Seite 158 (→ Kap. 4.4.5, [Projekt] / [Objekt] / [Einfügen] ([Project] / [Object] / [Add])).

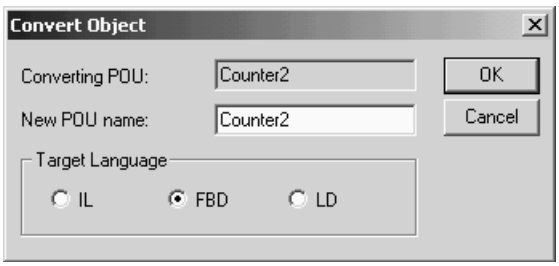
Schließen Sie den Dialog mit [OK].

4.4.7 [Projekt] / [Objekt] / [Konvertieren] ([Project] / [Object] / [Convert])

Der Menübefehl [Projekt] / [Objekt] / [Konvertieren] kann nur auf Bausteine angewandt werden. Dazu muss das Projekt übersetzt sein.

Sie können Bausteine, die in den Programmiersprachen ST, FUP, KOP und AWL geschrieben sind, in eine andere Programmiersprache (AWL, FUP oder KOP) konvertieren.

Nach dem Ausführen des Menübefehls wird der Dialog [Objekt konvertieren] ([Convert Object]) geöffnet.



20296AEN

Geben sie im Eingabefeld [Neuer Bausteinname] ([New POU name]) einen neuen Namen für den Baustein ein.

Wählen Sie unter [Zielsprache] ([Target language]) die Programmiersprache, in die Sie konvertieren möchten.

Hinweise:

Wenn Sie den Namen des Bausteins nicht ändern, wird der alte Bausteine überschrieben.

Bausteine mit Aktionen können nicht konvertiert werden.

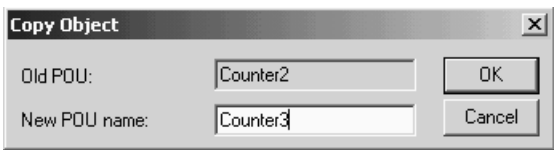
Einen Baustein, der in FUP programmiert ist, können Sie über den Menübefehl [Extras] / [Ansicht] ([Extras] / [View]) auch im KOP-Editor darstellen und bearbeiten, ohne dass Sie vorher eine Konvertierung durchführen müssen. Dies ist sowohl im Online-Modus als auch im Offline-Modus möglich.

Bestätigen Sie den Dialog mit [OK]. Der neue Baustein wird im Object Organizer eingefügt.

4.4.8 [Projekt] / [Objekt] / [Kopieren] ([Project] / [Object] / [Copy])

Mit dem Menübefehl [Projekt] / [Objekt] / [Kopieren] wird ein ausgewähltes Objekt kopiert und unter neuem Namen abgespeichert.

Nach dem Ausführen des Menübefehls [Projekt] / [Objekt] / [Kopieren] wird der Dialog [Objekt kopieren] ([Copy Object]) geöffnet.



20297AEN

Geben Sie im Eingabefeld [Name neuer Baustein] ([New POU name]) den Namen des neuen Objekts ein. Der Name des Objekts darf noch nicht für ein anderes Objekt verwendet worden sein. Ausnahme: Gleichnamige Aktionen sind erlaubt, wenn sie verschiedenen Objekten zugeordnet sind.

Hinweis:

Der Menübefehl [Projekt] / [Objekt] / [Kopieren] ist nicht funktionsgleich mit dem Menübefehl [Bearbeiten] / [Kopieren] ([Edit] / [Copy]), der das markierte Objekt in die Zwischenablage kopiert, ohne dass ein Dialog erscheint.

4.4.9 [Projekt] / [Objekt] / [Bearbeiten] ([Project] / [Object] / [Edit])

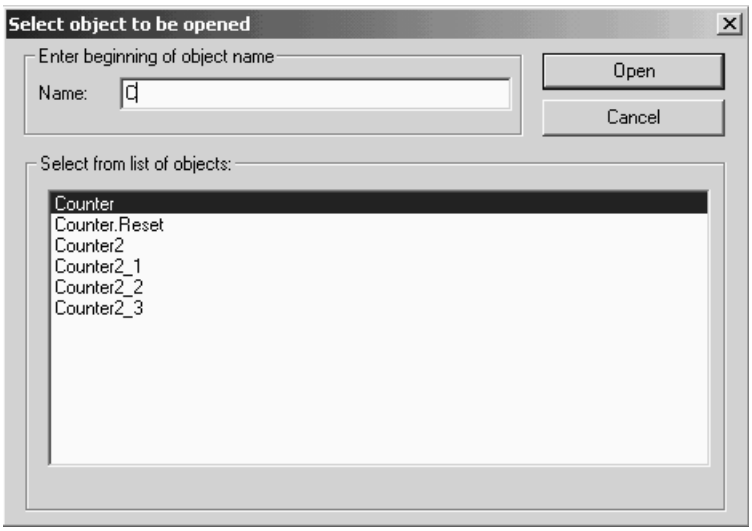
Mit dem Menübefehl [Projekt] / [Objekt] / [Bearbeiten] (oder <Einfügen>-Taste) laden Sie ein im **Object Organizer** markiertes Objekt in den Editor.

Wenn bereits ein Editorfenster mit diesem Objekt geöffnet ist, wird das Fenster in den Vordergrund geholt und und das Objekt kann bearbeitet werden.

Es gibt noch zwei weitere Möglichkeiten, um ein Objekt zum Bearbeiten zu öffnen:

- Doppelklicken Sie auf das gewünschte Objekt
- Geben Sie im Object Organizer die ersten Buchstaben des Objektnamens ein.

  Daraufhin wird der Dialog [Wähle zu öffnendes Objekt] ([Select object to be opened]), in dem alle Objekte der eingestellten Objektart mit diesen Anfangsbuchstaben zur Auswahl aufgeführt werden.



20298AEN

Aktionen werden in der Notation <Bausteinname>.<Aktionsname> aufgeführt. Da der Objektauswahldialog die Objekte alphabetisch listet, werden die Aktionen eines Bausteins immer diesem nachfolgend in der Liste aufgeführt.

Markieren Sie das gewünschte Element in der Liste und klicken Sie auf die Schaltfläche [Öffnen] ([Open]). Das Objekt wird im Object Organizer markiert und im Editor geöffnet.

Diese Vorgehensweise wird bei der Objektart "Ressourcen" nur für globale Variablen unterstützt.

4.4.10 [Projekt] / [Objekt] / [Eigenschaften] ([Project] / [Object] / [Properties])

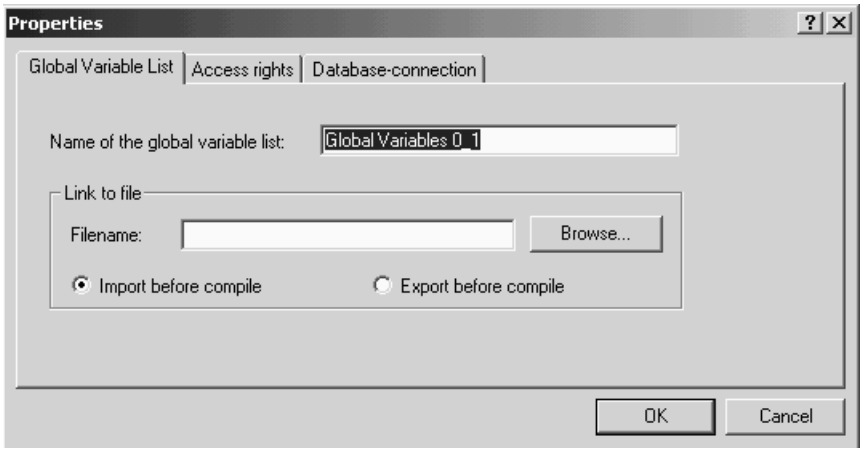
Nach dem Ausführen des Menübefehls [Projekt] / [Objekt] / [Eigenschaften] wird der Dialog [Eigenschaften] ([Properties]) angezeigt. Anzahl und Bezeichnung der Registerblätter sind vom Objekttyp abhängig und davon, ob das Projekt mit einer Projektdatenbank verbunden ist.

Insgesamt sind drei Registerblätter möglich:

- [Globale Variablenliste] ([Global Variable List])
- [Projektdatenbank] ([Database-connection])
- [Zugriffsrechte] ([Access rights])

Globale Variablenliste

Im Registerblatt [Globale Variablenliste] ([Global Variable List]) wird der Name der globalen Variablenliste angezeigt.



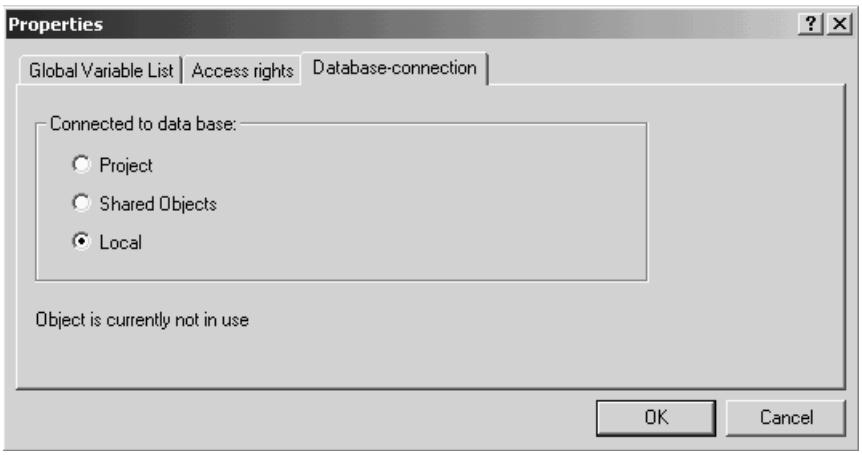
20585AEN

Wenn die globale Variablenliste in einer externen Datei enthalten ist, wird im Eingabefeld [Dateiname] ([Filename]) der Dateiname angezeigt. Über die Optionen [Vor Übersetzen importieren] und [Vor Übersetzen exportieren] ([Import before compile] und [Export before compile]) legen Sie fest, wie die globale Variablenliste beim Übersetzungsvorgang behandelt wird.

Weitere Informationen finden Sie auf Seite 277 (→ Kap. 6.2.3, Globale Variablen).

Projektdatenbank

Wenn das Projekt mit einer ENI-Projektdatenbank verknüpft ist, steht für jedes Objekt zusätzlich das Registerblatt [Projektdatenbank] ([Database-connection]) zur Verfügung.



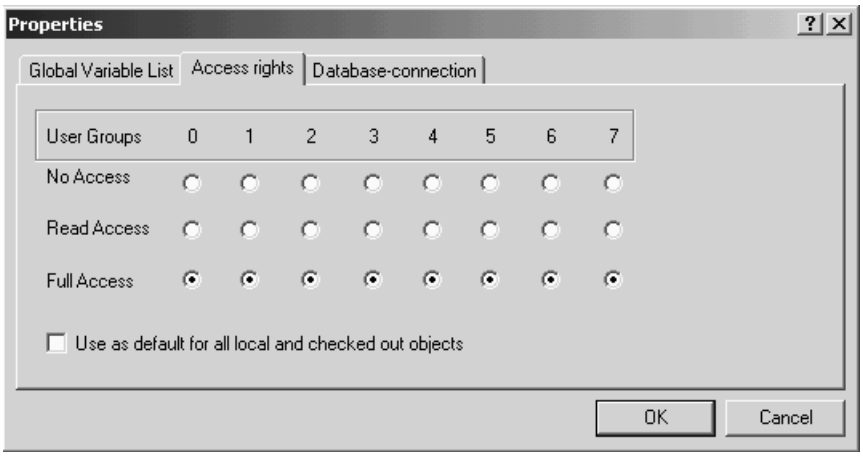
20586AEN

Informationen dazu, wie Sie ein Projekt mit einer Projektdatenbank verbinden, finden Sie auf Seite 93 (→ Kap. 4.2.13, [Projektdatenbank] ([Database-connection])).

In dem Registerblatt wird die aktuelle Zuordnung des Objekts zu einer der Datenbankkategorien [Projekt] oder [Gemeinsame Objekte] ([Project] oder [Shared Objects]) bzw. zur Kategorie [Lokal] ([Local]) angezeigt. Die Zuordnung kann in dem Registerblatt verändert werden.

Zugriffsrechte

In dem Registerblatt [Zugriffsrechte] ([Access rights]) werden die Zugriffsrechte der verschiedenen Arbeitsgruppen angezeigt und festgelegt.



20587AEN

Nur Mitglieder der Arbeitsgruppe 0 können die Zugriffsrechte ändern.

Für jede Arbeitsgruppe sind drei Einstellungen möglich:

- [Kein Zugriff] ([No Access])

  Das Objekt kann von Mitgliedern der Arbeitsgruppe nicht geöffnet werden.
- [Lesezugriff] ([Read Access])

  Das Objekt kann von Mitgliedern der Arbeitsgruppe geöffnet, aber nicht geändert werden.
- [Vollzugriff] ([Full Access])

  Das Objekt kann von Mitgliedern der Arbeitsgruppe geöffnet und geändert werden.

Wenn Sie die Option [Für alle lokalen und ausgecheckten Objekte übernehmen] ([Use as default for all local and checked out objects]) aktivieren, werden die Einstellungen nicht für das im Object Organizer markierte Objekt, sondern für alle lokalen und ausgecheckten Objekte übernommen. Die Einstellungen betreffen dann also alle Bausteine, Datentypen und Ressourcen.

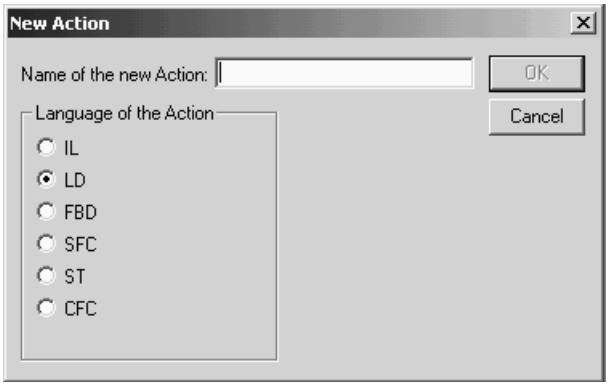
Informationen zu Arbeitsgruppen und der Passwortvergabe finden Sie auf Seite 144 (→ Kap. 4.3.26, Arbeitsgruppen).

4.4.11 [Projekt] / [Aktion hinzufügen] ([Project] / [Add Action])

Mit dem Menübefehl [Projekt] / [Aktion hinzufügen] erstellen Sie für einen Baustein eine neue Aktion.

Markieren Sie dazu den Baustein im **Object Organizer** und führen Sie den Menübefehl [Projekt] / [Aktion hinzufügen] aus.

Der Dialog [Neue Aktion] ([New Action]) wird geöffnet.



20588AEN

Geben Sie einen Namen für die Aktion ein und legen Sie die Programmiersprache fest.

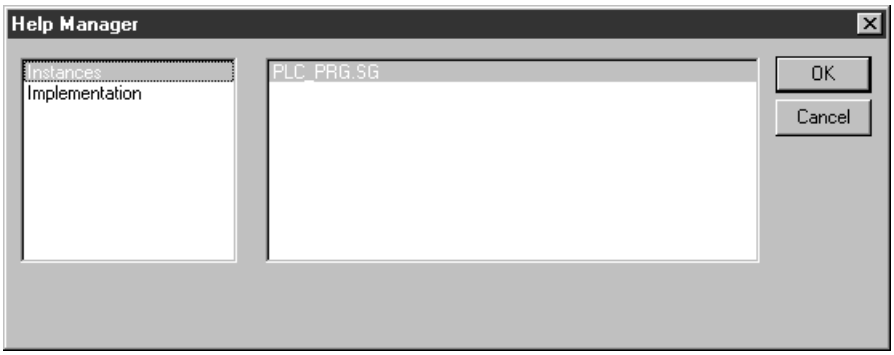
Die neue Aktion wird im Object Organizer unter den Baustein gehängt und im Editor geöffnet.

Weitere Informationen zu Aktionen finden Sie auf Seite 22 (→ Abschnitt. Aktion) und auf Seite 33 (→ Abschnitt. Aktion).

4.4.12 [Projekt] / [Instanz öffnen] ([Project] / [View Instance])

Mit dem Menübefehl [Projekt] / [Instanz öffnen] können Sie im Online-Modus die Instanz des im Object Organizer ausgewählten Funktionsblock im Editor anzeigen lassen.

Markieren Sie den Funktionsblock und führen Sie den Menübefehl [Projekt] / [Instanz öffnen] aus oder doppelklicken Sie auf den Funktionsblock. Der Dialog [Eingabehilfe] ([Help Manager]) wird geöffnet.



20589AEN

In dem Dialog sind die Instanzen und die Implementationen des Funktionsblocks aufgelistet. Markieren Sie eine Instanz bzw. Implementation und bestätigen Sie mit [OK]. Die Instanz oder Implementation wird im Editor geöffnet.

Hinweis:

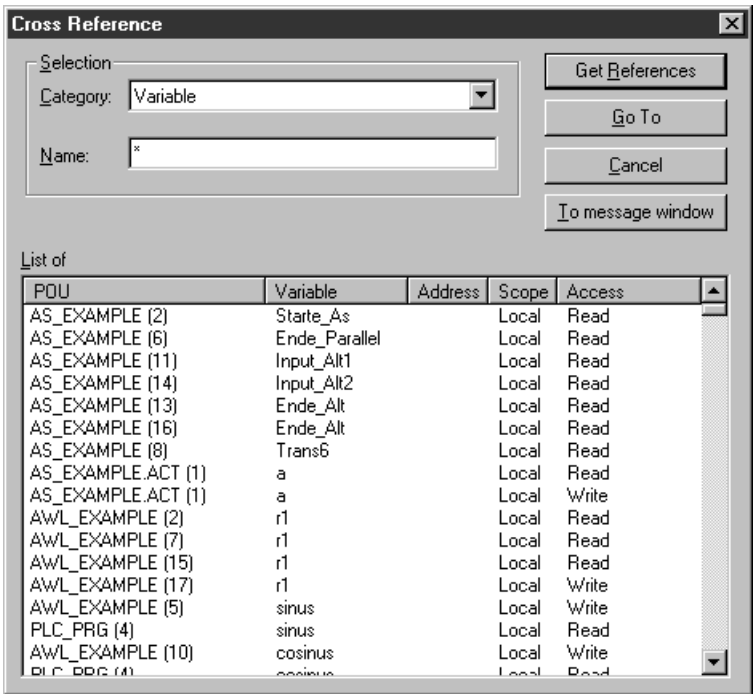
Instanzen können erst nach dem Einloggen geöffnet werden!

4.4.13 [Projekt] / [Querverweisliste ausgeben] ([Project] / [Show Cross Reference])

Hinweis:

Das Projekt muss übersetzt sein, damit der Menübefehl korrekt ausgeführt wird.

Nach dem Ausführen des Menübefehls [Projekt] / [Querverweisliste ausgeben] wird der Dialog [Querverweise] ([Cross Reference]) geöffnet.


Cross Reference
Selection
Category: Variable
Name: *
Get References
Go To
Cancel
To message window
List of
POU | Variable | Address | Scope | Access
AS_EXAMPLE (2) | Starte_As | | Local | Read
AS_EXAMPLE (6) | Ende_Parallel | | Local | Read
AS_EXAMPLE (11) | Input_Alt1 | | Local | Read
AS_EXAMPLE (14) | Input_Alt2 | | Local | Read
AS_EXAMPLE (13) | Ende_Alt | | Local | Read
AS_EXAMPLE (16) | Ende_Alt | | Local | Read
AS_EXAMPLE (8) | Trans6 | | Local | Read
AS_EXAMPLE.ACT (1) | a | | Local | Read
AS_EXAMPLE.ACT (1) | a | | Local | Write
AWL_EXAMPLE (2) | r1 | | Local | Read
AWL_EXAMPLE (7) | r1 | | Local | Read
AWL_EXAMPLE (15) | r1 | | Local | Read
AWL_EXAMPLE (17) | r1 | | Local | Write
AWL_EXAMPLE (5) | sinus | | Local | Write
PLC_PRG (4) | sinus | | Local | Read
AWL_EXAMPLE (10) | cosinus | | Local | Write

20590AEN

Falls das Projekt seit dem letzten Übersetzen geändert wurde, erscheint in der Titelzeile des Dialogs der Hinweis "(Nicht aktuell)". Damit auch neu hinzugekommene Querverweise angezeigt werden, müssen Sie das Projekt erneut übersetzen.

In dem Dialog sind alle Verwendungsstellen einer Variable, einer Adresse oder eines Bausteins aufgelistet.

Wählen Sie zuerst die Objektkategorie aus:

- Variable
- Adresse
- Baustein

Geben Sie anschließend den Namen des Objekts an. Alternativ können Sie über die <F2>-Taste den Dialog [Eingabehilfe] ([Help Manager]) öffnen und dort ein Objekt auswählen.

Um die Querverweise aller Elemente einer Kategorie anzuzeigen, geben Sie bei [Name] ([Name]) ein Sternchen "*" ein.

Klicken Sie auf die Schaltfläche [Querverweise] ([Get References]), um die Querverweisliste zu erstellen.

Folgende Informationen werden jeweils dargestellt:

- Bausteinname mit Zeilen- bzw. Netzwerknummer
- Variablenname
- Adresse (wenn vorhanden)
- Gültigkeitsbereich (lokal oder global)
- Zugriffsart auf das Objekt (lesend oder schreibend)

Sie können sich die Verwendungsstellen im Editor anzeigen lassen:

Markieren Sie dazu einen Listeneintrag und klicken Sie auf die Schaltfläche [Gehe zu] ([Go to]) oder doppelklicken Sie auf einen Listeneintrag.

Der Baustein, in dem das Objekt verwendet wird, wird im Editor geöffnet und alle Verwendungsstellen des Objekts markiert.

Alternativ können Sie die Querverweisliste über die Schaltfläche [Ins Meldungsfenster] ([To message window]) in das Meldungsfenster übernehmen.

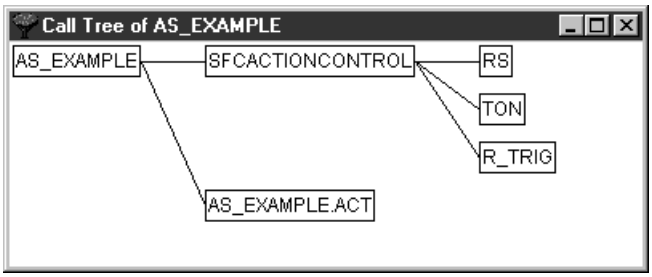
Doppelklicken Sie anschließend im Meldungsfenster auf einen Listeneintrag, um den Baustein im Editor zu öffnen.

4.4.14 [Projekt] / [Aufrufbaum ausgeben] ([Project] / [Show call tree])

Hinweis:

Das Projekt muss übersetzt sein, damit der Menübefehl korrekt ausgeführt wird.

Nach dem Ausführen des Menübefehls [Projekt] / [Aufrufbaum ausgeben] wird ein Fenster geöffnet, in dem der Aufrufbaum des im **Object Organizer** markierten Objekts dargestellt ist. Der Aufrufbaum zeigt, welche Bausteine im Objekt aufgerufen werden.



20591AEN

4.5 Allgemeine Editierfunktionen

4.5.1 Übersicht

Die in diesem Kapitel beschriebenen Menübefehle stehen in allen Editoren und zum Teil im **Object Organizer** zur Verfügung. Die Menübefehle befinden sich im Menü [Bearbeiten] ([Edit]) und im Kontextmenü.

Wenn die IntelliPoint-Software auf dem Computer installiert ist, unterstützt der PLC-Editor die Funktionen des Rades und der Radtaste der Microsoft IntelliMouse.

Die IntelliMouse können Sie in allen Editoren für die Zoomfunktion benutzen: Halten Sie zum Vergrößern die <Strg>-Taste gedrückt und drehen Sie das Mausrad vorwärts. Zum Verkleinern drehen Sie das Mausrad rückwärts.

Die Menübefehle finden Sie in folgenden Kapiteln:

- (→ Kap. 4.5.2, [Bearbeiten] / [Rückgängig] ([Edit] / [Undo])), Seite 167
- (→ Kap. 4.5.3, [Bearbeiten] / [Wiederherstellen] ([Edit] / [Redo])), Seite 168
- (→ Kap. 4.5.4, [Bearbeiten] / [Ausschneiden] ([Edit] / [Cut])), Seite 168
- (→ Kap. 4.5.5, [Bearbeiten] / [Kopieren] ([Edit] / [Copy])), Seite 169
- (→ Kap. 4.5.6, [Bearbeiten] / [Einfügen] ([Edit] / [Paste])), Seite 169
- (→ Kap. 4.5.7, [Bearbeiten] / [Löschen] ([Edit] / [Delete])), Seite 170
- (→ Kap. 4.5.8, [Bearbeiten] / [Suchen] ([Edit] / [Find])), Seite 170
- (→ Kap. 4.5.9, [Bearbeiten] / [Weitersuchen] ([Edit] / [Find next])), Seite 171
- (→ Kap. 4.5.10, [Bearbeiten] / [Ersetzen] ([Edit] / [Replace])), Seite 172
- (→ Kap. 4.5.11, [Bearbeiten] / [Eingabehilfe] ([Edit] / [Input Assistant])), Seite 173
- (→ Kap. 4.5.12, [Bearbeiten] / [Variablen Deklaration] ([Edit] / [Auto declare])), Seite 176
- (→ Kap. 4.5.13, [Bearbeiten] / [Nächster Fehler] ([Edit] / [Next Error])), Seite 176
- (→ Kap. 4.5.14, [Bearbeiten] / [Vorheriger Fehler] ([Edit] / [Previous Error])), Seite 177
- (→ Kap. 4.5.15, [Bearbeiten] / [Makros] ([Edit] / [Macros])), Seite 177

4.5.2 [Bearbeiten] / [Rückgängig] ([Edit] / [Undo])

Tastaturbefehl: <Strg>+<Z>

Der Menübefehl [Bearbeiten] / [Rückgängig] macht im aktuell geöffneten Editorfenster bzw. im **Object Organizer** nacheinander alle seit dem letzten Öffnen des Fensters ausgeführten Aktionen – beginnend mit der letzten Aktion – rückgängig.

Mit dem Menübefehl [Bearbeiten] / [Wiederherstellen] ([Edit] / [Redo]) können Sie eine rückgängig gemachte Aktion wiederherstellen.

Hinweis:

Die Menübefehle [Rückgängig] und [Wiederherstellen] beziehen sich jeweils auf das aktuelle Fenster. Jedes Fenster führt seine eigene Aktionsliste. Wenn Sie in mehreren Fenstern Aktionen rückgängig machen wollen, aktivieren Sie jeweils das entsprechende Fenster. Gleiches gilt für den Object Organizer.

4.5.3 [Bearbeiten] / [Wiederherstellen] ([Edit] / [Redo])

Tastaturbefehl: <Strg>+<Y>

Mit dem Menübefehl [Bearbeiten] / [Wiederherstellen] können Sie eine im aktuellen Editorfenster bzw. im **Object Organizer** rückgängig gemachte Aktion wiederherstellen.

Hinweis:

Die Menübefehle [Rückgängig] und [Wiederherstellen] ([Undo] und [Redo]) beziehen sich jeweils auf das aktuelle Fenster. Jedes Fenster führt seine eigene Aktionsliste. Wenn Sie in mehreren Fenstern Aktionen rückgängig machen wollen, aktivieren Sie jeweils das entsprechende Fenster. Gleiches gilt für den Object Organizer.

4.5.4 [Bearbeiten] / [Ausschneiden] ([Edit] / [Cut])

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<X> oder <Umschalt>+<Entf>

In einem Editorfenster kopiert der Menübefehl [Bearbeiten] / [Ausschneiden] den Inhalt der Markierung in die Zwischenablage und entfernt anschließend den markierten Inhalt.

Beim **Object Organizer** gilt dies analog für das markierte Objekt. Allerdings lassen sich nicht alle Objekte löschen, z. B. die Steuerungskonfiguration.

Hinweis:

Nicht alle Editoren unterstützen das Ausschneiden, in einigen ist es nur eingeschränkt möglich.

Der markierte Inhalt hängt vom Editor ab:

- In den Texteditoren AWL und ST und in den Deklarationen ist die Markierung eine Liste von Zeichen.
- In den grafischen Editoren FUP und KOP ist die Markierung eine Menge von Netzwerken, die jeweils durch ein gepunktetes Rechteck im Netzwerk-Zahlenfeld markiert ist, bzw. eine Box mit allen vorangehenden Linien, Boxen und Operanden.
- Im AS-Editor ist die Auswahl ein Teil einer Schrittfolge, umgeben von einem gepunkteten Rechteck.

Um den Inhalt der Zwischenablage einzufügen, benutzen Sie den Menübefehl [Bearbeiten] / [Einfügen] ([Edit] / [Paste]).

Im AS-Editor können Sie außerdem folgende Menübefehle benutzen:

- [Extras] / [Parallelzweig einfügen (rechts)] ([Extras] / [Paste Parallel Branch (right)])
- [Extras] / [Einfügen danach] ([Extras] / [Paste after])

Um eine Auswahl in die Zwischenablage zu kopieren, ohne sie zu entfernen, benutzen Sie den Menübefehl [Bearbeiten] / [Kopieren] ([Edit] / [Copy]).

Um einen markierten Bereich zu entfernen, ohne ihn in die Zwischenablage zu kopieren, benutzen Sie den Menübefehl [Bearbeiten] / [Löschen] ([Edit] / [Delete]).

4.5.5 [Bearbeiten] / [Kopieren] ([Edit] / [Copy])

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<C>

Der Menübefehl [Bearbeiten] / [Kopieren] kopiert im Editorfenster den Inhalt der aktuellen Markierung in die Zwischenablage. Der Inhalt des Editorfensters wird dabei nicht verändert.

Beim **Object Organizer** gilt dies analog für das markierte Objekt. Allerdings lassen sich nicht alle Objekte kopieren, z. B. die Steuerungskonfiguration.

Hinweis:

Nicht alle Editoren unterstützen das Kopieren, in einigen ist es nur eingeschränkt möglich.

Der markierte Inhalt hängt vom Editor ab:

- In den Texteditoren AWL und ST und in den Deklarationen ist die Markierung eine Liste von Zeichen.
- In den grafischen Editoren FUP und KOP ist die Markierung eine Menge von Netzwerken, die jeweils durch ein gepunktetes Rechteck im Netzwerk-Zahlenfeld markiert ist, bzw. eine Box mit allen vorangehenden Linien, Boxen und Operanden.
- Im AS-Editor ist die Auswahl ein Teil einer Schrittfolge, umgeben von einem gepunkteten Rechteck.

Um den Inhalt der Zwischenablage einzufügen, benutzen Sie den Menübefehl [Bearbeiten] / [Einfügen] ([Edit] / [Paste]).

Im AS-Editor können Sie außerdem folgende Menübefehle benutzen:

- [Extras] / [Parallelzweig einfügen (rechts)] ([Extras] / [Paste Parallel Branch (right)])
- [Extras] / [Einfügen danach] ([Extras] / [Paste after])

Um einen markierten Bereich in die Zwischenablage zu kopieren und gleichzeitig zu entfernen, benutzen Sie den Menübefehl [Bearbeiten] / [Ausschneiden] ([Edit] / [Cut]).

4.5.6 [Bearbeiten] / [Einfügen] ([Edit] / [Paste])

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<V>

Der Menübefehl [Bearbeiten] / [Einfügen] fügt den Inhalt der Zwischenablage an die aktuelle Position im Editorfenster ein.

In den grafischen Editoren ist der Menübefehl nur ausführbar, wenn durch das Einfügen wieder eine korrekte Struktur entsteht.

Beim **Object Organizer** wird das Objekt aus der Zwischenablage eingefügt.

Die aktuelle Position wird je nach Editor unterschiedlich definiert:

- Bei den Texteditoren AWL und ST und in den Deklarationen ist die aktuelle Position durch den blinkenden Cursor markiert. Der Cursor ist ein senkrechter Strich, der mit der Maus oder den Pfeiltasten positioniert werden kann.
- In den grafischen Editoren FUP und KOP ist die aktuelle Position das erste Netzwerk mit einem gepunkteten Rechteck im Netzwerknummernbereich. Der Inhalt der Zwischenablage wird vor diesem Netzwerk eingefügt. Wenn eine Teilstruktur eingefügt werden soll, wird diese vor dem markierten Element eingefügt.

- Im AS-Editor ist die aktuelle Position durch die Auswahl festgelegt, die mit einem gepunkteten Rechteck umgeben ist. Der Inhalt der Zwischenablage wird, abhängig von der Markierung und dem Inhalt der Zwischenablage, vor dieser Markierung oder in einem neuen Zweig (parallel oder alternativ) links der Markierung eingefügt.

Im AS-Editor können Sie außerdem folgende Menübefehle verwenden:

- [Extras] / [Parallelzweig einfügen (rechts)] ([Extras] / [Paste Parallel Branch (right)])
- [Extras] / [Einfügen danach] ([Extras] / [Paste after])

Um einen markierten Bereich in die Zwischenablage zu kopieren ohne ihn zu entfernen, benutzen Sie den Menübefehl [Bearbeiten] / [Kopieren] ([Edit] / [Copy]).

Um einen markierten Bereich in die Zwischenablage zu kopieren und gleichzeitig zu entfernen, benutzen Sie den Menübefehl [Bearbeiten] / [Ausschneiden] ([Edit] / [Cut]).

4.5.7 [Bearbeiten] / [Löschen] ([Edit] / [Delete])

Tastaturbefehl: <Entf>

Der Menübefehl [Bearbeiten] / [Löschen] löscht den markierten Bereich im Editorfenster.

Beim **Object Organizer** gilt dies analog für das markierte Objekt. Allerdings lassen sich nicht alle Objekte löschen, z. B. die Steuerungskonfiguration.

Im Bibiotheksverwalter gilt dies für den markierten Bibliotheksnamen.

Der markierte Bereich hängt vom Editor ab:

- In den Texteditoren AWL und ST und in den Deklarationen ist die Markierung eine Liste von Zeichen.
- In den grafischen Editoren FUP und KOP ist die Markierung eine Menge von Netzwerken, die jeweils durch ein gepunktetes Rechteck im Netzwerk-Zahlenfeld markiert ist, bzw. eine Box mit allen vorangehenden Linien, Boxen und Operanden.
- Im AS-Editor ist die Auswahl ein Teil einer Schrittfolge, umgeben von einem gepunkteten Rechteck.

Um einen markierten Bereich zu entfernen und gleichzeitig in die Zwischenablage zu kopieren, benutzen Sie den Menübefehl [Bearbeiten] / [Ausschneiden] ([Edit] / [Cut]).

4.5.8 [Bearbeiten] / [Suchen] ([Edit] / [Find])

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<F>

Der Menübefehl [Bearbeiten] / [Suchen] öffnet den Dialog [Suchen] ([Find]).

20596AEN

Sie können im aktuellen Editorfenster nach Textstellen suchen. Der Dialog bleibt so lange geöffnet, bis Sie ihn mit [Abbrechen] ([Cancel]) beenden.

Im Eingabefeld [Suchen nach] ([Find what]) wird automatisch die Zeichenfolge angezeigt, die im Editor markiert ist. Sie können auch eine Zeichenfolge eingeben, die Sie suchen möchten. Die zuletzt gesuchten Zeichenfolgen können Sie über die Liste des Eingabefeldes [Suchen nach] auswählen.

Ihnen stehen zwei Optionen zur Verfügung, um die Anzahl der Suchergebnisse einschränken:

- [Nur ganzes Wort suchen] ([Match whole word only])

  Wenn diese Option akiviert ist, werden nur Suchergebnisse angezeigt, die exakt der gesuchten Zeichenfolge entsprechen.

  Wenn diese Option deaktiviert ist, werden auch Suchergebnisse angezeigt, die die gesuchte Zeichenfolge enthalten.

  Beispiel: Die gesuchte Zeichenfolge ist "Auto". Als Suchergebnisse werden sowohl "Auto" als auch "Automat" angezeigt.

- [Groß-/Kleinschreibung] ([Match case])

  Wenn diese Option akitiviert ist, werden nur Suchergebnisse angezeigt, bei denen die Groß-/Kleinschreibung identisch ist mit der gesuchten Zeichenfolge.

  Wenn diese Option deaktiviert ist, wird die Groß-/Kleinschreibung ignoriert.

  Beispiel: Die gesuchte Zeichenfolge ist "Auto". Als Suchergebnisse werden sowohl "Auto" als auch "auto" angezeigt.

Sie können die Suchrichtung innerhalb des Editorfensters festlegen, ausgehend von der aktuellen Cursorposition:

- [Nach oben] ([Up])
- [Nach unten] ([Down])

Im CFC-Editor wird dabei die geometrische Anordnung der Elemente von links oben nach rechts unten berücksichtigt. In FUP-Bausteinen erfolgt die Abarbeitung von rechts nach links.

Um den Suchvorgang zu starten, klicken Sie auf die Schaltfläche [Weitersuchen] ([Find next]). Wenn die Zeichenfolge gefunden wurde, wird die Textstelle im Editor markiert. Wiederholen Sie den Suchvorgang gegebenenfalls, um weitere Fundstellen für die Zeichenfolge zu finden.

Wenn die Suche erfolglos war oder keine weiteren Treffer gefunden wurden, wird eine entsprechende Meldung ausgegeben.

Hinweis:

Beachten Sie, dass die gefundene Textstelle vom Suchdialog verdeckt sein kann. Verschieben Sie bei Bedarf das Dialogfenster.

4.5.9 [Bearbeiten] / [Weitersuchen] ([Edit] / [Find next])

Schaltfläche:

Tastaturbefehl: <F3>

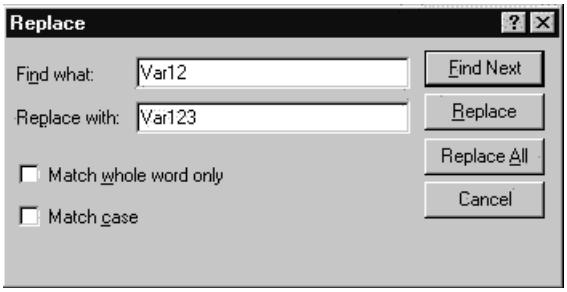
Mit dem Menübefehl [Bearbeiten] / [Weitersuchen] wiederholen Sie den letzten Suchbefehl mit denselben Einstellungen, die Sie vorher unter dem Menübefehl [Bearbeiten] / [Suchen] ([Edit] / [Find]) vorgenommen haben.

Beachten Sie, dass in FUP-Bausteinen die Abarbeitung von rechts nach links erfolgt.

4.5.10 [Bearbeiten] / [Ersetzen] ([Edit] / [Replace])

Tastaturbefehl: <Strg>+<H>

Der Menübefehl [Bearbeiten] / [Ersetzen] öffnet den Dialog [Ersetzen] ([Replace]).



20598AEN

Sie können im aktuellen Editorfenster nach Textstellen suchen, analog zum Menübefehl [Bearbeiten] / [Suchen] ([Edit] / [Find]). Zusätzlich haben Sie hier noch die Möglichkeit, den gefundenen Text durch einen anderen zu ersetzen.

Der Dialog bleibt so lange geöffnet, bis Sie ihn mit [Abbrechen] ([Cancel]) beenden.

Im Eingabefeld [Suchen nach] ([Find what]) wird automatisch die Zeichenfolge angezeigt, die im Editor markiert ist. Sie können auch eine Zeichenfolge eingeben, die Sie suchen möchten. Die zuletzt gesuchten Zeichenfolgen können Sie über die Liste des Eingabefeldes [Suchen nach] auswählen.

Im Eingabefeld [Ersetzen durch] ([Replace with]) geben Sie den Text ein, mit dem der gefundene Text ersetzt werden soll.

Ihnen stehen zwei Optionen zur Verfügung, um die Anzahl der Suchergebnisse einschränken:

- [Nur ganzes Wort suchen] ([Match whole word only])

  Wenn diese Option akiviert ist, werden nur Suchergebnisse angezeigt, die genau der gesuchten Zeichenfolge entsprechen.

  Wenn diese Option deaktiviert ist, werden auch Suchergebnisse angezeigt, die die gesuchte Zeichenfolge enthalten.

  Beispiel: Die gesuchte Zeichenfolge ist "Auto". Als Suchergebnisse werden sowohl "Auto" als auch "Automat" angezeigt.

- [Groß-/Kleinschreibung] ([Match case])

  Wenn diese Option akitiviert ist, werden nur Suchergebnisse angezeigt, bei denen die Groß-/Kleinschreibung identisch ist mit der gesuchten Zeichenfolge.

  Wenn diese Option deaktiviert ist, wird die Groß-/Kleinschreibung ignoriert.

  Beispiel: Die gesuchte Zeichenfolge ist "Auto". Als Suchergebnisse werden sowohl "Auto" als auch "auto" angezeigt.

Um den Suchvorgang zu starten, klicken Sie auf die Schaltfläche [Weitersuchen] ([Find next]), . Wenn die Zeichenfolge gefunden wurde, wird die Textstelle im Editor markiert, der Text aber noch nicht ersetzt.

Wenn Sie den gefundenen Text ersetzen möchten, klicken Sie auf die Schaltfläche [Ersetzen] ([Replace]).

Wiederholen Sie den Vorgang, um den Suchtext an anderen Stellen zu finden und zu ersetzen.

Wenn die gesuchte Zeichenfolge an mehreren Stellen auftaucht und an allen Stellen ersetzt werden soll, können Sie dies über die Schaltfläche [Alles ersetzen] ([Replace all]) in einem Arbeitsschritt durchführen.

Wenn die Suche erfolglos war oder keine weiteren Treffer gefunden wurden, wird eine entsprechende Meldung ausgegeben.

Hinweise:

Gehen Sie mit dem Befehl [Alles ersetzen] vorsichtig um, damit Sie nicht ungewollt an der falschen Stelle Text ersetzen.

Schreibgeschützter Text kann nicht ersetzt werden: Teile der Task- und Steuerungskonfiguration, Bibliotheken. Zeichenfolgen in editierbaren Teilen der Konfigurationen können ersetzt werden: Task- und Programmname, Bezeichner für Ein- und Ausgänge.

Beachten Sie, dass die gefundene Textstelle vom Suchdialog verdeckt sein kann. Verschieben Sie bei Bedarf das Dialogfenster.

4.5.11 [Bearbeiten] / [Eingabehilfe] ([Edit] / [Input Assistant])

Tastaturbefehl: <F2>

Nach dem Ausführen des Menübefehls [Bearbeiten] / [Eingabehilfe] wird der Dialog [Eingabehilfe] ([Help Manager]) geöffnet. Dieser Dialog zeigt alle Eingaben, die an der aktuellen Cursorposition möglich sind.

Wählen Sie im linken Fenster die gewünschte Kategorie aus und doppelkicken Sie anschließend auf einen Eintrag im rechten Fenster. Sie können den Eintrag auch auswählen, indem Sie ihn durch einen einfachen Mausklick markieren und mit [OK] bestätigen.

Der ausgewählte Eintrag wird daraufhin an der Cursorposition eingefügt.

In Abhängigkeit vom Editor und von der Cursorposition, werden in der Eingabehilfe alle Kategorien und Einträge angezeigt, die an dieser Position erlaubt sind: Variablen, Operatoren, Bausteine, Konvertierungen usw.

Argumente

Wenn der ausgewählte Eintrag ein Element mit Argumenten ist, z. B. ein Funktionsblock, der eine Eingangsvariable hat, erscheint unterhalb des linken Fensters die zusätzliche Option [Mit Argumenten] ([With arguments]).

Wenn Sie diese Option aktivieren, wird das Element mit den zu übergebenden Argumenten eingefügt, z. B. `func1(var1, var2);`.

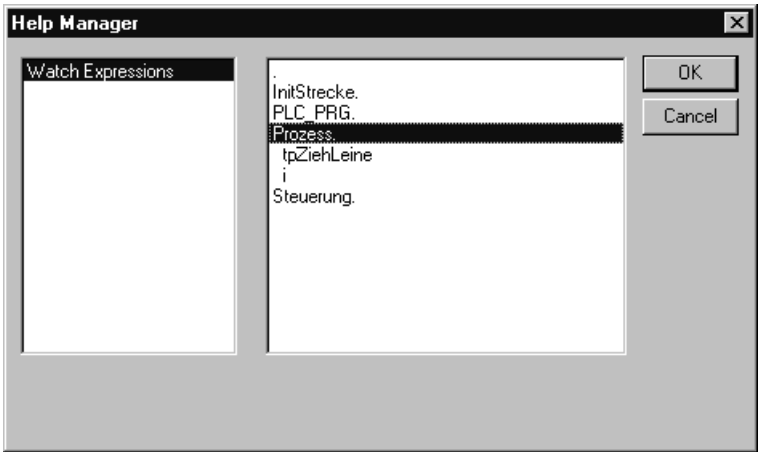
Darstellung

Für die meisten Kategorien steht die Option [Strukturierte Darstellung] ([Structured]) zur Verfügung, die unterhalb des rechten Fensters erscheint.

Durch Aktivieren bzw. Deaktivieren dieser Option wechseln Sie zwischen strukturierter und nicht strukturierter Darstellung im rechten Fenster.

Nicht strukturierte Darstellung

In dieser Darstellung werden Bausteine, Variablen und Datentypen in einer einfachen, alphabetisch sortierten Liste dargestellt.



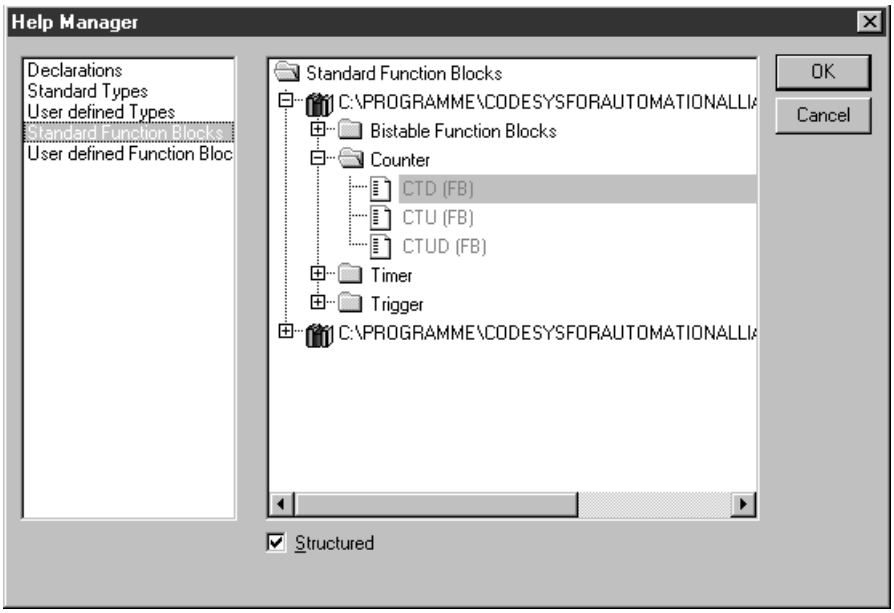
20599AEN

An manchen Positionen, z. B. in der Watchliste, werden mehrstufige Variablennamen benötigt. In diesem Fall werden im rechten Fenster eine Liste aller Bausteine und ein einzelner Punkt für die globalen Variablen angezeigt.

Hinter jedem Bausteinnamen ist ebenfalls ein Punkt. Wenn Sie auf einen Baustein doppelklicken, wird die Liste der zugehörige Variablen geöffnet. Wenn Instanzen und Datentypen vorliegen, können Sie weitere Listen aufklappen. Um die Variable zu übernehmen, doppelklicken Sie auf die Variable oder bestätigen Sie mit [OK].

Strukturierte Darstellung

In der strukturierten Darstellung werden Bausteine, Variablen oder Datentypen hierarchisch angeordnet.



20498AEN

Die strukturierte Darstellung ist für folgende Kategorien möglich:

- Standard-Programme
- Standard-Funktionen
- Standard-Funktionsblöcke
- Definierte Programme
- Definierte Funktionen
- Definierte Funktionsblöcke
- Globale Variablen
- Lokale Variablen
- Definierte Typen
- Watch-Variablen

Darstellungsweise und Hierarchie entsprechen denen des **Object Organizer**. Elemente aus Bibliotheken werden in alphabetischer Reihenfolge an oberster Stelle eingefügt und die Hierarchie wie im Bibliotheksverwalter dargestellt.

Ein- und Ausgangsvariablen von Funktionsblöcken, die als lokale oder globale Variablen deklariert sind, werden in den Kategorien [Lokale Variablen] bzw. [Globale Variablen] ([Local Variables] bzw. [Global Variables]) unterhalb des Instanznamens aufgelistet, z. B. `Inst_TP.ET`, `Inst_TP.IN`. Um zu den Variablen zu gelangen, doppelklicken Sie auf den Instanznamen, z. B. `Inst_TP`.

Wenn die Instanz eines Funktionsblocks markiert ist, können Sie die Option [Mit Argumenten] (With Arguments]) aktivieren. Dann werden in den Textsprachen ST und AWL und bei der Taskkonfiguration außer dem Instanznamen auch die Eingangsparameter des Funktionsblocks eingefügt, z. B. `Inst(IN:= , PT:=`.

Bei den grafischen Programmiersprachen und im Watch-Fenster wird generell nur der Instanzname eingefügt.

Komponenten von Strukturen werden analog zu Funktionsblockinstanzen dargestellt.

Für Enumerationen werden die einzelnen Enumerationswerte unter dem Enumerationstyp in folgender Reihenfolge aufgelistet:

- ENUM aus Bibliotheken
- ENUM aus Datentypen
- lokale ENUM aus Bausteinen

Generell gilt, dass Zeilen, die Unterobjekte enthalten, nicht ausgewählt, sondern nur auf- und zugeklappt werden können. Einzige Ausnahme davon bilden Instanzen.

Watch- und Rezepturverwalter, Tracekonfiguration

Wenn Sie die Eingabehilfe im Watch- und Rezepturverwalter bzw. in der Tracekonfiguration aufrufen, können sie mehrere Einträge gleichzeitig auswählen. Benutzen Sie dazu die <Umschalt>-Taste oder die <Strg>-Taste plus Mausklick. Wenn Sie bei der Mehrfachauswahl versuchen, Einträge auszuwählen, die keine gültigen Variablen enthalten, so werden diese Einträge nicht in die Auswahl übernommen. Diese Einträge können Sie auch nicht einzeln markieren.

Im Watchfenster und in der Tracekonfiguration können Sie Strukturen, Arrays oder Instanzen aus der Eingabehilfe übernehmen. Allerdings können Sie diese Einträge nicht durch Doppelklick übernehmen. Sie müssen die Übernahme immer mit [OK] bestätigen. Nachdem Sie mit [OK] bestätigt haben, werden die Variablen zeilenweise ins Watchfenster eingetragen, wobei jede Variable in eine Zeile geschrieben wird. Bei den Tracevariablen wird jede Variable in einer Zeile der Tracevariablen-Liste eingetragen.

Hinweis:

In der Tracevariablen-Liste sind max. 20 Einträge erlaubt. Wenn bei der Übernahme von Travevariablen aus der Eingabehilfe diese Grenze erreicht wird, erscheint die Meldung "Es sind höchstens 20 Variablen erlaubt". Weitere ausgewählte Variablen werden dann nicht mehr in die Liste übernommen.

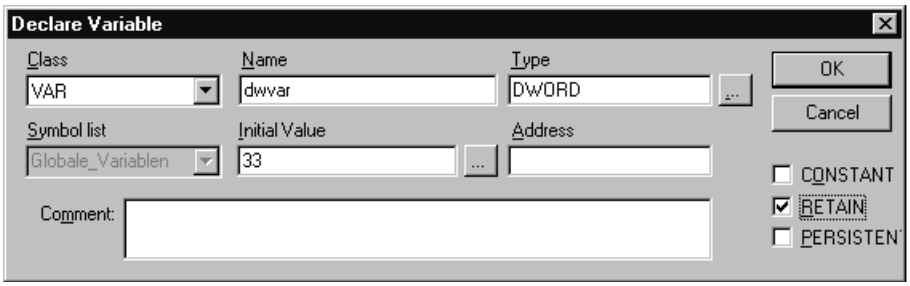
Hinweis:

Einige Einträge in der Eingabehilfe, z. B. globale Variablen, werden erst nach einem Übersetzungslauf aktualisiert.

4.5.12 [Bearbeiten] / [Variablen Deklaration] ([Edit] / [Auto declare])

Tastaturbefehl: <Umschalt>+<F2>

Nach dem Ausführen des Menübefehls [Bearbeiten] / [Variablen Deklaration] wird der Dialog [Variablendeklaration] ([Declare Variable]) geöffnet.



20499AEN

Dieser Dialog wird bei der Deklaration einer neuen Variablen automatisch geöffnet, wenn Sie in den Projektoptionen, Kategorie [Editor] ([Editor]), die Option [Automatisch deklarieren] ([Autodeclaration]) akivieren.

Eine ausführliche Beschreibung der Einstellmöglichkeiten finden Sie auf Seite 207 (→ Kap. 5.2.17, Automatisch deklarieren).

4.5.13 [Bearbeiten] / [Nächster Fehler] ([Edit] / [Next Error])

Tastaturbefehl: <F4>

Wenn ein Übersetzungslauf nicht fehlerfrei durchgeführt wurde, können Sie sich mit dem Menübefehl [Bearbeiten] / [Nächster Fehler] den nächsten Fehler bzw. die nächste Warnung anzeigen lassen.

Nach dem Ausführen des Menübefehls wird der entsprechende Editor geöffnet und die fehlerhafte Stelle markiert. Gleichzeitig wird die Markierung im Meldungsfenster auf den nächsten Eintrag gesetzt.

Wenn Sie nur Fehlermeldungen, aber keine Warnungen durchlaufen möchten, müssen Sie in den Projektoptionen, Kategorie [Arbeitsbereich] ([Desktop]) die Option [F4 ignoriert Warnungen] ([F4 ignores warnings]) aktivieren.

Siehe dazu auch Seite 177 (→ Kap. 4.5.14, [Bearbeiten] / [Vorheriger Fehler] ([Edit] / [Previous Error])).

4.5.14 [Bearbeiten] / [Vorheriger Fehler] ([Edit] / [Previous Error])

Tastaturbefehl: <Umschalt>+<F4>

Wenn ein Übersetzungslauf nicht fehlerfrei durchgeführt wurde, können Sie sich mit dem Menübefehl [Bearbeiten] / [Vorheriger Fehler] den vorherigen Fehler bzw. die vorherige Warnung anzeigen lassen.

Nach dem Ausführen des Menübefehls wird der entsprechende Editor geöffnet und die fehlerhafte Stelle markiert. Gleichzeitig wird die Markierung im Meldungsfenster vom aktuellen Eintrag auf den Eintrag davor gesetzt.

Wenn Sie nur Fehlermeldungen, aber keine Warnungen durchlaufen möchten, müssen Sie in den Projektoptionen, Kategorie [Arbeitsbereich] ([Desktop]) die Option [F4 ignoriert Warnungen] ([F4 ignores warnings]) aktivieren.

Siehe dazu auch Seite 176 (→ Kap. 4.5.13, [Bearbeiten] / [Nächster Fehler] ([Edit] / [Next Error])).

4.5.15 [Bearbeiten] / [Makros] ([Edit] / [Macros])

Unter dem Menüpunkt [Bearbeiten] / [Makros] erscheint ein Untermenü mit allen Makros, die für das aktuelle Projekt definiert wurden. Wenn Sie ein Makro anklicken und das Makro ausführbar ist, wird der Dialog [Makro ausführen] ([Execute Makro]) geöffnet. Hier erscheinen der Makroname und die aktuelle Befehlszeile. Über die Schaltfäche [Abbrechen] ([Cancel]) können Sie die Abarbeitung des Makros stoppen, wobei die aktuelle Befehlszeile noch zu Ende abgearbeitet wird. Nach einem Abbruch wird folgende Meldung ausgegeben: "<Makroname>: Ausführung durch Anwender abgebrochen". Wenn Sie sich im Online-Modus befinden, wird die Meldung zusätzlich in das Logbuch eingetragen.

Makros können im Offline-Modus und im Online-Modus ausgeführt werden. Es werden jedoch nur die im jeweiligen Modus verfügbaren Befehle ausgeführt.

4.6 Allgemeine Online-Funktionen

4.6.1 Einleitung

Die Online-Befehle finden Sie unter dem Menüpunkt [Online] ([Online]). Die Ausführung einiger Online-Befehle ist abhängig vom aktiven Editor.

Die Online-Befehle stehen erst nach dem Einloggen zur Verfügung.

Die Funktion **Online Change** ermöglicht Ihnen, Änderungen des Programms auf der laufenden Steuerung vorzunehmen.

Weitere Informationen zu Online Change finden Sie auf Seite 178 (→ Kap. 4.6.2, [Online] / [Einloggen] ([Online] / [Login])).

4.6.2 [Online] / [Einloggen] ([Online] / [Login])

Schaltfläche: []

Tastaturbefehl: <Alt>+<F8>

Über den Menübefehl [Online] / [Einloggen] verbinden Sie das Programmiersystem mit der Steuerung (bzw. starten das Simulationsprogramm) und wechseln in den Online-Modus.

Wenn das aktuelle Projekt seit dem Öffnen bzw. seit der letzten Veränderung nicht übersetzt wurde, wird es beim Einloggen übersetzt. Wenn beim Übersetzen Fehler auftreten, wechselt der PLC-Editor nicht in den Online-Modus.

Weitere Informationen zum Übersetzen finden Sie auf Seite 122 (→ Kap. 4.3.11, [Projekt] / [Übersetzen] ([Project] / [Build])).

Nach dem erfolgreichen Einloggen stehen alle Onlinefunktionen zur Verfügung, wenn die entsprechenden Einstellungen in den Projektoptionen, Kategorie [Übersetzungsoptionen] ([Build]) vorgenommen wurden.

Um vom Online-Modus wieder in den Offline-Modus zu wechseln, benutzen Sie den Menübefehl [Online] / [Ausloggen] ([Online] / [Logout]).

Systemmeldungen beim Einloggen

"Das gewählte Steuerungsprofil entspricht nicht dem des Zielsystems ..."

Prüfen Sie, ob die Zielsystemeinstellungen im **Object Organizer**, Kategorie [Ressourcen] ([Resources]) mit den unter dem Menübefehl [Online] / [Kommunikationsparameter] ([Online] / [Communication Parameters]) eingestellten Parametern übereinstimmt.

"Kommunikationsfehler. Es wird ausgeloggt."

Prüfen Sie, ob die Steuerung läuft.

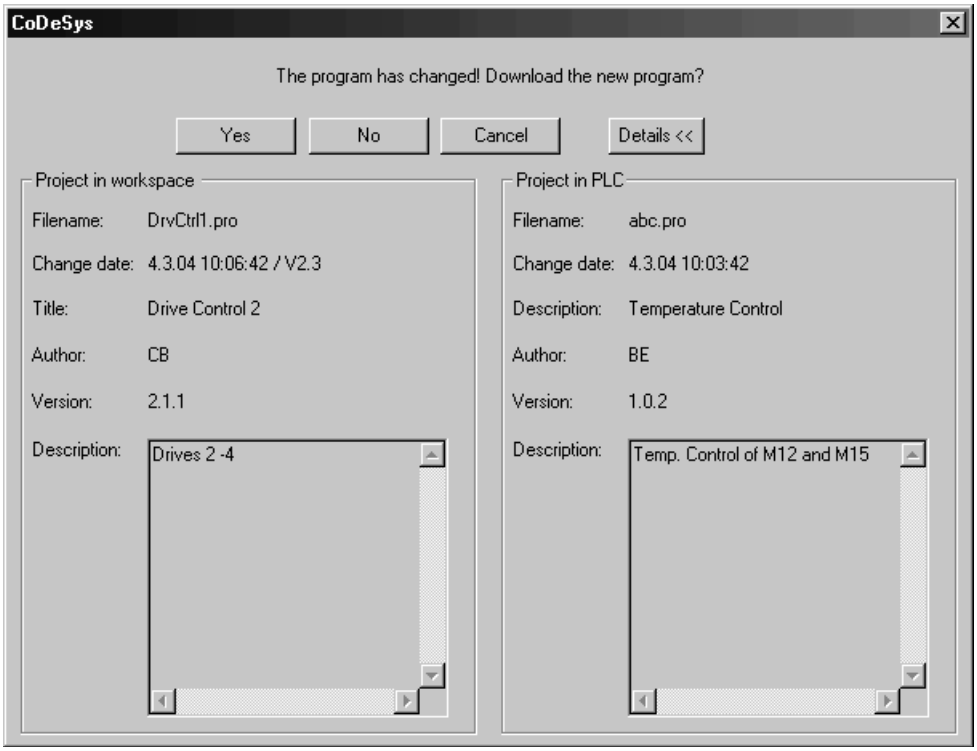
Prüfen Sie, ob die unter dem Menübefehl [Online] / [Kommunikationsparameter] ([Online] / [Communication Parameters]) eingestellten Parameter mit denen in der Steuerung übereinstimmen. Prüfen Sie insbesondere, ob der richtige Port eingestellt ist und ob die Baudraten in Steuerung und Programmiersystem übereinstimmen.

Wenn Sie einen Gateway-Server benutzen, prüfen Sie, ob der richtige Kanal eingestellt ist.

"Das Programm wurde geändert! Soll das neue Programm geladen werden?" bzw.
"Das Programm wurde geändert! Sollen die Änderungen geladen werden? (ONLINE CHANGE)"

Das aktuelle Projekt im PLC-Editor passt nicht zu dem derzeit in der Steuerung gelade-

nen. **Monitoring** und **Debugging** sind deshalb nicht möglich.



20549AEN

Online Change

Wenn in den Projektoptionen, Kategorie [Arbeitsbereich] ([Desktop]) die Option [Online-Betrieb im Sicherheitsmodus] ([Online in security mode]) aktiviert ist und das Zielsystem diese Funktion unterstützt, werden in dem Dialog automatisch Details zum Status des Projekts im PLC-Editor und in der Steuerung angezeigt. Die Projektinformationen können über die Schaltfläche [Details] ([Details]) ein- und ausgeblendet werden.

Wenn die Option [Online-Betrieb im Sicherheitsmodus] nicht aktiviert ist, können Sie die Details über die Schaltfläche [Details] manuell öffnen.

Die Fehlermeldung hat bei Online Change folgende Bedeutung:

Das Projekt läuft auf der Steuerung. Das Zielsystem unterstützt Online Change und das Projekt ist im Vergleich zum letzten Download bzw. dem letzten Online Change verändert worden.

Sie können nun entscheiden, ob diese Änderungen bei laufendem Steuerungsprogramm geladen werden sollen oder ob der Befehl abgebrochen werden soll. Sie können aber auch den gesamten übersetzten Code laden, indem Sie die Schaltfläche [Alles laden] ([Load all]) wählen.

Wenn Sie mit [Ja] ([Yes]) bestätigen, werden beim Einloggen die geänderten Teile des Projekts auf die Steuerung geladen. Sehen Sie hierzu auch Seite 180 (→ Abschnitt. Hinweise zu Online Change), .

Wenn Sie auf die Schaltfläche [Nein] ([No]) klicken, erfolgt das Einloggen, ohne dass die seit dem letzten Download vorgenommenen Änderungen auf die Steuerung geladen werden.

Mit [Abbrechen] ([Cancel]) brechen Sie den Menübefehl [Online] / [Einloggen] ab.

Bitte beachten Sie, dass es von den Einstellungen im Zielsystem abhängt, welche Schaltfläche automatisch markiert ist, also den Fokus hat.

Hinweise zu Online Change

- Online Change ist nicht möglich:
  - nach Änderungen in der Taskkonfiguration
  - nach Änderungen in der Steuerungskonfiguration
  - nach dem Einfügen einer Bibliothek
  - nach dem Menübefehl [Projekt] / [Alles bereinigen] ([Project] / [Clear all])
- Wenn die Download-Information (Datei `<Projektname><Targetidentifier>.ri`), die beim letzten Laden des Projekts oder beim letzten Online Change angelegt worden war, gelöscht wurde (beispielsweise über den Menübefehl [Projekt] / [Alles bereinigen] ([Project] / [Clear all]), ist kein Online Change mehr möglich.

  Wenn Sie von der Download-Information eine Kopie gespeichert haben, können Sie Menübefehl [Projekt] / [Download-Information laden] ([Project] / [Load download information]) wieder geladen werden.
- Beim Online Change werden die Werte nicht neu initialisiert. Änderungen der Initialisierungswerte werden nicht berücksichtigt!
- Retain-Variablen behalten beim Online Change ihre Werte, im Gegensatz zu einem erneuten Download des Projekts.

  Weitere Informationen hierzu finden Sie auf Seite 180 (→ Kap. 4.6.4, [Online] / [Laden] ([Online] / [Download])).

4.6.3 [Online] / [Ausloggen] ([Online] / [Logout])

Schaltfläche: []

Tastaturbefehl: <Strg>+<F8>

Die Verbindung zur Steuerung wird abgebaut bzw. das Simulationsprogramm beendet und in den Offline-Modus gewechselt.

Um wieder in den Online-Modus zu wechseln, verwenden Sie den Menübefehl [Online] / [Einloggen] ([Online] / [Login]).

4.6.4 [Online] / [Laden] ([Online] / [Download])

Mit dem Menübefehl [Online] / [Laden] laden Sie das übersetzte Projekt in die Steuerung.

Hinweis:

Der Menübefehl [Online] / [Laden] darf nicht verwechselt werden mit dem Menübefehl [Online] / [Quellcode laden] ([Online] / [Sourcecode download])!

Die Download-Information wird beim Übersetzen erzeugt. Wenn Sie die C-Code-Generierung benutzen, wird vor dem Laden der C-Compiler aufgerufen, der die Download-Datei erzeugt.

Die Download-Informationen werden in einer Datei `<projectname>0000000ar.ri` gespeichert. Diese Datei wird beim **Online Change** verwendet, um das aktuelle Projekt mit dem zuletzt in die Steuerung geladenen Projekt zu vergleichen. Beim Online Change werden dann nur die geänderten Programmteile erneut geladen.

Die Download-Information wird beim Ausführen des Menübefehls [Projekt] / [Alles bereinigen] ([Project] / [Clear all]) ebenfalls gelöscht!

Zielsystemabhängig können Sie bei jedem Erzeugen eines Bootprojekts im Offline-Modus die Download-Datei automatisch neu erzeugen lassen.

Hinweis:

Persistente Variablen behalten ihren Wert nach einem Download.

4.6.5 [Online] / [Start] ([Online] / [Start])

Schaltfläche: [ ]

Tastaturbefehl: <F5>

Mit dem Menübefehl [Online] / [Start] starten Sie die Abarbeitung des Projekts in der Steuerung bzw. in der Simulation.

Der Menübefehl kann nach bzw. bei folgenden Ereignissen ausgeführt werden:

- Unmittelbar nach dem Menübefehl [Online] / [Laden] ([Online] / [Download])
- Nach einem Stopp der Ausführung des Projekts in der Steuerung mit dem Menübefehl [Online] / [Stop] ([Online] / [Stop])
- Die Ausführung des Projekts steht an einen Breakpoint.
- Nach dem Menübefehl [Online] / [Einzelzyklus] ([Online] / [Single Cycle])

4.6.6 [Online] / [Stop] ([Online] / [Stop])

Schaltfläche: [ ]

Tastaturbefehl: <Umschalt>+<F8>

Der Menübefehl [Online] / [Stop] stoppt die Abarbeitung des Anwenderprogramms in der Steuerung bzw. in der Simulation zwischen zwei Zyklen.

Um die Programmabarbeitung fortzusetzen, verwenden Sie den Menübefehl [Online] / [Start] ([Online] / [Start]).

4.6.7 [Online] / [Reset] ([Online] / [Reset])

Der Menübefehl [Online] / [Reset] setzt mit Ausnahme der `RETAIN`-Variablen alle Variablen auf den Wert zurück, mit dem sie initialisiert wurden, also auch die mit `VAR PERSISTENT` deklarierten Variablen! Variablen, die nicht explizit mit einem Initialisierungswert versehen wurden, werden auf die Standardinitialwerte gesetzt, Integer-Zahlen beispielsweise auf "0". Bevor alle Variablen überschrieben werden, erfolgt eine Sicherheitsabfrage durch den PLC-Editor. Die Situation entspricht der bei einem Stromausfall oder beim Aus-/Einschalten der Steuerung (Warmstart), während das Programm läuft.

Um die Steuerung und damit die Programmabarbeitung erneut zu starten, verwenden Sie den Menübefehl [Online] / [Start] ([Online] / [Start]).

Weitere Informationen finden Sie in folgenden Kapiteln:

- (→ Kap. 4.6.8, [Online] / [Reset (kalt)] ([Online] / [Reset (cold)])), Seite 182
- (→ Kap. 4.6.9, [Online] / [Reset (Ursprung)] ([Online] / [Reset (original)])), Seite 182
- (→ Kap. 5.2.7, Remanente Variablen), Seite 202

4.6.8 [Online] / [Reset (kalt)] ([Online] / [Reset (cold)])

Der Menübefehl [Online] / [Reset (kalt)] entspricht dem Menübefehl [Online] / [Reset] ([Online] / [Reset]) mit dem Unterschied, dass alle Variablen – also auch die `RETAIN`-Variablen – auf ihren Initialisierungswert zurückgesetzt werden. Die Situation entspricht der beim Start eines Programms, das neu auf die Steuerung geladen wurde (Kaltstart).

Weitere Informationen finden Sie in folgenden Kapiteln:

- (→ Kap. 4.6.7, [Online] / [Reset] ([Online] / [Reset])), Seite 181
- (→ Kap. 4.6.9, [Online] / [Reset (Ursprung)] ([Online] / [Reset (original)])), Seite 182
- (→ Kap. 5.2.7, Remanente Variablen), Seite 202

4.6.9 [Online] / [Reset (Ursprung)] ([Online] / [Reset (original)])

Der Menübefehl [Online] / [Reset (Ursprung)] setzt alle Variablen – auch die remanenten Variablen `VAR RETAIN` und `VAR PERSISTENT` – auf ihren Initialisierungswert zurück und löscht das Anwenderprogramm auf der Steuerung. Die Steuerung wird in den Urzustand zurückversetzt.

Weitere Informationen finden Sie in folgenden Kapiteln:

- (→ Kap. 4.6.7, [Online] / [Reset] ([Online] / [Reset])), Seite 181
- (→ Kap. 4.6.8, [Online] / [Reset (kalt)] ([Online] / [Reset (cold)])), Seite 182
- (→ Kap. 5.2.7, Remanente Variablen), Seite 202

4.6.10 [Online] / [Breakpoint an/aus] ([Online] / [Toggle Breakpoint])

Schaltfläche: [ ]

Tastaturbefehl: <F9>

Die Position, an der ein Breakpoint gesetzt werden kann, hängt von der Sprache ab, in der der Baustein im aktiven Fenster geschrieben ist.

In den Texteditoren AWL und ST wird der Breakpoint auf die Zeile gesetzt, in der der Cursor steht, wenn diese Zeile eine Breakpoint-Position ist. Eine Breakpoint-Position erkennen Sie an der dunkelgrauen Farbe des Zeilennummernfeldes (bei Standardeinstellung). Um in den Texteditoren einen Breakpoint zu setzen bzw. zu entfernen, können Sie auch auf das Zeilennummernfeld klicken.

In den grafischen Editoren FUP und KOP wird der Breakpoint auf das aktuell markierte Netzwerk gesetzt. Um in den grafischen Editoren einen Breakpoint zu setzen bzw. zu entfernen, können Sie auch auf das Netzwerknummernfeld klicken.

Im AS-Editor wird der Breakpoint auf den aktuell markierten Schritt gesetzt. Um im AS-Editor einen Breakpoint zu setzen bzw. zu entfernen, können Sie auch die <Umschalt>-Taste plus Doppelklick verwenden.

Wenn ein Breakpoint gesetzt ist, werden das Zeilennummernfeld, das Netzwerknummernfeld und der Schritt mit hellblauer Hintergrundfarbe dargestellt (bei Standardeinstellung).

Wenn bei der Programmabarbeitung ein Breakpoint erreicht ist, stoppt das Programm und das entsprechende Feld wird mit einer roten Hintergrundfarbe dargestellt (bei Standardeinstellung).

Um das Programm fortzusetzen, verwenden Sie einen der folgenden Menübefehle:

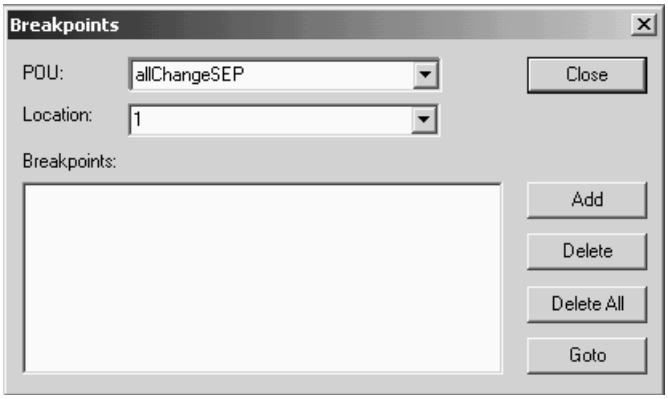
- [Online] / [Start] ([Online] / [Start])
- [Online] / [Einzelschritt in] ([Online] / [Step in])
- [Online] / [Einzelschritt über] ([Online] / [Step over])

Zum Setzen bzw. Entfernen von Breakpoints können Sie auch den Menübefehl [Online] / [Breakpoint-Dialog] ([Online] / [Breakpoint Dialog]) verwenden.

4.6.11 [Online] / [Breakpoint-Dialog] ([Online] / [Breakpoint Dialog])

Der Menübefehl [Online] / [Breakpoint-Dialog] ermöglicht das Editieren der Breakpoints des gesamten Projekts.

Nach dem Ausführen des Menübefehls wird der Dialog [Breakpoints] ([Breakpoints]) geöffnet.



20751AEN

Der Dialog zeigt alle aktuell gesetzten Breakpoints an.

Zum Setzen eines Breakpoints wählen Sie in der Liste [Baustein] ([POU]) einen Baustein und in der Liste [Ort] ([Location]) die Zeile bzw. das Netzwerk, wo der Breakpoint gesetzt werden soll. Drücken Sie anschließend die Schaltfläche [Hinzufügen] ([Add]). Der Breakpoint wird in die Liste aufgenommen.

Um einen Breakpoint zu löschen, markieren Sie den Breakpoint, den Sie löschen möchten und klicken Sie auf die Schaltfläche [Löschen] ([Delete]).

Mit der Schaltfläche [Alle löschen] ([Delete all]) werden alle Breakpoints gelöscht.

Um im Editor zu der Stelle zu gehen, an der ein bestimmter Breakpoint gesetzt wurde, markieren Sie den Breakpoint und drücken Sie die Schaltfläche [Gehe zu] ([Go to]).

Um Breakpoints zu setzen bzw. entfernen, können Sie auch den Menübefehl [Online] / [Breakpoint an/aus] ([Online] / [Toggle Breakpoint]) verwenden.

4.6.12 [Online] / [Einzelschritt über] ([Online] / [Step over])

Schaltfläche: [ ]

Tastaturbefehl: <F10>

Mit dem Menübefehl [Online] / [Einzelschritt über] wird ein Einzelschritt ausgeführt. Die Abarbeitung eines Bausteins wird dabei als Einzelschritt behandelt. Im AS-Editor wird eine komplette Aktion als Einzelschritt abgearbeitet.

Wenn die aktuelle Anweisung der Aufruf einer Funktion oder eines Funktionsblocks ist, dann wird die Funktion oder der Funktionsblock komplett ausgeführt. Um die Abarbeitu ng an der ersten Anweisung einer aufgerufenen Funktion bzw. eines aufgerufenen Funktionsblocks zu stoppen, verwenden Sie den Menübefehl [Online] / [Einzelschritt in] ([Online] / [Step in].

Wenn die letzte Anweisung der Funktion oder des Funktionsblocks erreicht ist, kehrt das Programm zum aufrufenden Baustein zurück und geht zur nächsten Anweisung dieses Bausteins.

4.6.13 [Online] / [Einzelschritt in] ([Online] / [Step in])

Tastaturbefehl: <F8>

Mit dem Menübefehl [Online] / [Einzelschritt in] wird ein Einzelschritt abgearbeitet. Wenn dabei ein Baustein aufgerufen wird, stoppt die Abarbeitung vor Ausführung der ersten Anweisung des Bausteins.

Gegebenenfalls wird in einen aufgerufenen Baustein gewechselt.

Wenn die aktuelle Position ein Aufruf einer Funktion oder eines Funktionsblocks ist, wechselt das Programm zur ersten Anweisung des aufgerufenen Bausteins.

In allen anderen Situationen verhält sich der Menübefehl wie der Menübefehl [Online] / [Einzelschritt über] ([Online] / [Step over]).

4.6.14 [Online] / [Einzelzyklus] ([Online] / [Single Cycle])

Tastaturbefehl: <Strg>+<F5>

Der Menübefehl [Online] / [Einzelzyklus] führt einen einzelnen Steuerungszyklus aus und stoppt nach diesem Steuerungszyklus.

Der Menübefehl kann kontinuierlich wiederholt werden, um in einzelnen Steuerungszyklen fortzufahren.

Um den Einzelzyklus abzubrechen, verwende Sie den Menübefehl [Online] / [Start] ([Online] / [Start]).

4.6.15 [Online] / [Werte schreiben] ([Online] / [Write Values])

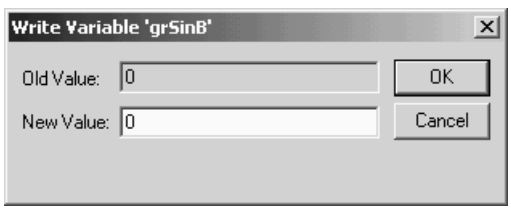
Tastaturbefehl: <Strg>+<F7>

Mit dem Menübefehl [Online] / [Werte schreiben] werden zu Beginn eines Zyklus – einmalig – Variablen auf benutzerdefinierte Werte gesetzt.

Sie können die Werte aller einelementigen Variablen verändern, die auch im Monitoring sichtbar sind.

Bevor Sie den Menübefehl ausführen können, müssen Sie einen Variablenwert zum Schreiben vorbereiten:

- Nicht-boolesche Variablen
  Doppelklicken Sie auf die Zeile, in der die Variable deklariert ist, oder markieren Sie die Variable und drücken Sie die <Eingabetaste>-Taste.
  Daraufhin wird der Dialog [Variable <x> schreiben] ([Write Variable <x>]) geöffnet. Geben Sie im Eingabefeld [Neuer Wert] ([New Value]) den Wert ein, der auf die Variable geschrieben werden soll.



20753AEN

- Boolesche Variablen
  Um den Wert der Variablen zu wechseln, doppelklicken Sie auf die Zeile, in der die Variable deklariert ist.

  Der zum Schreiben vorgesehene neue Wert wird türkisfarben in spitzen Klammern hinter dem bisherigen Deklarationswert angezeigt:

20754AXX

Sie können für beliebig viele Variablen Werte setzen.

Die Werte, die für Variablen zum Schreiben eingetragen wurden, können auf die gleiche Weise auch korrigiert bzw. wieder gelöscht werden.

Hinweis:

Ausnahme bei der Anzeige der zu schreibenden Werte: In Editoren für FUP und KOP steht der Wert ohne spitze Klammern neben dem Variablennamen.

Hinweis:

Das Schreiben von Variablenwerten ist auch über den Menübefehl [Online] / [Schreiben/Forcen-Dialog] ([Online] / [Write/Force-Dialog]) möglich.

Die zum Schreiben vorgemerkten Werte werden in der **Watchliste** gespeichert. In der Watchliste bleiben die Werte, bis sie tatsächlich geschrieben, gelöscht oder durch den Menübefehl [Online] / [Werte forcen] ([Online] / [Force Values]) in die **Forceliste** verschoben werden.

Watch- und Forceliste können Sie über den Menübefehl [Online] / [Schreiben/Forcen-Dialog] ([Online] / [Write/Force-Dialog]) einsehen.

Wenn der Menübefehl [Online] / [Werte schreiben] ausgeführt wird, werden alle in der Schreibliste enthaltenen Werte einmalig am Beginn eines Zyklus auf die entsprechenden Variablen in der Steuerung geschrieben und die Werte aus der Schreibliste gelöscht.

Hinweis:

In der Ablaufsprache AS können die Einzelwerte eines Transitionsausdrucks nicht über den Menübefehl [Online] / [Werte schreiben] verändert werden. Der Grund ist, dass beim Monitoring der "Gesamtwert" des Transitionsausdrucks dargestellt wird und nicht die Einzelvariablen.

Beispiel: a `AND` b wird nur dann als `TRUE` dargestellt, wenn beide Variablen den Wert `TRUE` haben.

Im Funktionsplan FUP ist bei einem Ausdruck, der beispielsweise als Eingang eines Funktionsblocks verwendet wird, nur das Schreiben der ersten Variable möglich. Der Grund ist, dass beim Monitoring nur die erste Variable dargestellt wird.

4.6.16 [Online] / [Werte forcen] ([Online] / [Force Values])

Tastaturbefehl: <F7>

Mit dem Menübefehl [Online] / [Werte forcen] werden eine oder mehrere Variablen dauerhaft auf benutzerdefinierte Werte gesetzt. Das Setzen erfolgt im Laufzeitsystem jeweils am Anfang und am Ende eines Zyklus.

Zeitlicher Ablauf eines Zyklus:

- 1. Eingänge lesen
- 2. Werte forcen
- 3. Code abarbeiten,
- 4. Werte forcen
- 5. Ausgänge schreiben

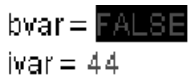
Der Menübefehl ist so lange aktiv, bis Sie den Menübefehl [Online] / [Forcen aufheben] ([Online] / [Release Force]) ausführen oder das Programmiersystem ausloggt.

Zum Setzen der neuen Werte wird zunächst eine Schreibliste (**Watchliste**) erzeugt. Die

in der Watchliste enthaltenen Variablen sind im **Monitoring** entsprechend gekennzeichnet. Nach dem Ausführen des Menübefehls [Online] / [Werte forcen] werden die Werte aus der Watchliste in die **Forceliste** übertragen.

Watch- und Forceliste können Sie über den Menübefehl [Online] / [Schreiben/Forcen-Dialog] ([Online] / [Write/Force-Dialog]) einsehen.

Die Watchliste wird geleert und die neuen Werte werden rot als "forced" gekennzeichnet; z. B.:



20755AXX

Modifikationen in der Forceliste werden jeweils beim nächsten Ausführen des Menübefehls [Online] / [Werte forcen] an das Programm übertragen.

Hinweis:

Die Forceliste entsteht beim ersten Ausführen des Menübefehls [Online] / [Werte forcen], während die Watchliste bereits vor dem ersten Schreiben der enthaltenen Variablen existiert.

Hinweis:

Wenn die Verbindung zur Steuerung z. B. durch Ausloggen unterbrochen wird, wird in der Regel die Forceliste auf der Steuerung gelöscht. Bei einigen Zielsystemen bleibt die Forceliste jedoch erhalten.

Das "Forcen" einer Variable und damit die Aufnahme des Wertes in die Forceliste ist auch über den Menübefehl [Online] / [Schreiben/Forcen-Dialog] ([Online] / [Write/Force-Dialog]) möglich.

Hinweis:

In der Ablaufsprache AS können Sie die Einzelwerte eines Transitionsausdrucks nicht über den Menübefehl [Online] / [Werte schreiben] verändern. Der Grund ist, dass beim Monitoring der "Gesamtwert" des Transitionsausdrucks dargestellt wird und nicht die Einzelvariablen.

Beispiel: a `AND` b wird nur dann als `TRUE` dargestellt, wenn beide Variablen den Wert `TRUE` haben.

Im Funktionsplan FUP ist bei einem Ausdruck, der beispielsweise als Eingang eines Funktionsblocks verwendet wird, nur das Schreiben der ersten Variable möglich. Der Grund ist, dass beim Monitoring nur die erste Variable dargestellt wird.

4.6.17 [Online] / [Forcen aufheben] ([Online] / [Release Force])

Tastaturbefehl: <Umschalt>+<F7>

Mit dem Menübefehl [Online] / [Forcen aufheben] beenden Sie das Forcing von Variablenwerten in der Steuerung. Die Variablen werden im **Monitoring** nicht mehr rot dargestellt und ändern ihren Wert wieder normal. Die **Forceliste** wird gelöscht.

Wenn die Schreibliste leer ist, wird der Menübefehl [Online] / [Forcen aufheben] direkt ausgeführt. Informationen zur Schreibliste finden Sie auf Seite 184 (→ Kap. 4.6.15, [Online] / [Werte schreiben] ([Online] / [Write Values])).

Wenn die Schreibliste nicht leer ist, wird nach dem Ausführen des Menübefehls der Dialog [Löschen der Schreib-/Forcelisten] ([Remove Write-/Forcelists]) geöffnet.

20760AEN

Wenn Sie in dem Dialog die Option [Schreibliste verwerfen] ([Remove writelist]) aktivieren, wird mit dem Menübefehl auch die Schreibliste gelöscht.

Sie haben die Möglichkeit, die komplette Forceliste zu löschen oder einzelne Variablen vor dem Ausführen des Menübefehls [Online] / [Forcen aufheben] für das Löschen vorzumerken.

Außer mit dem Menübefehl [Online] / [Forcen aufheben], können Sie die komplette Forceliste auch auf eine der folgenden Arten löschen:

- Führen Sie den Menübefehl [Online] / [Schreiben/Forcen-Dialog] ([Online] / [Write/Force-Dialog]) aus. Der Dialog [Editieren der Schreibliste und der Forceliste] ([Editing the writelist and the forcelist]) wird geöffnet. Klicken Sie in dem Dialog auf die Schaltfläche [Forcen aufheben] ([Release Force]).
- Löschen der kompletten Forceliste über den Dialog [Löschen der Schreib-/Forcelisten] (siehe unten). Dieser erscheint beim Befehl [Online] / [Forcen aufheben].

Um das Forcing nur für einzelne Variablen aufzuheben, müssen Sie diese Variablen zunächst dafür vormerken.

Wählen Sie dafür eine der folgenden Möglichkeiten. Die zum Forcen vorgemerkten Variablen sind danach am türkisfarbenen Zusatz <Forcen aufheben> erkenntlich:

- Ein doppelter Mausklick auf auf eine Zeile, in der eine nicht-boolesche Variable deklariert ist, öffnet den Dialog [Variable <x> schreiben]. Drücken Sie hier die Schaltfläche [Entfernen] ([Remove]).
- Bei booleschen Variablen doppelklicken Sie solange auf die Zeile, bis die türkisfarbenen Angaben hinter der Variable verschwindet.
- Löschen Sie im Dialog [Online] / [Schreiben/Forcen-Dialog] ([Online] / [Write/Force-Dialog]), Registerkarte [Forceliste] ([Forcelist]) den Wert in der Spalte [Geforcter Wert] ([Forced value]). Weitere Informationen dazu finden Sie auf Seite 187 (→ Kap. 4.6.18, [Online] / [Schreiben/Forcen-Dialog] ([Online] / [Write/Force-Dialog])).

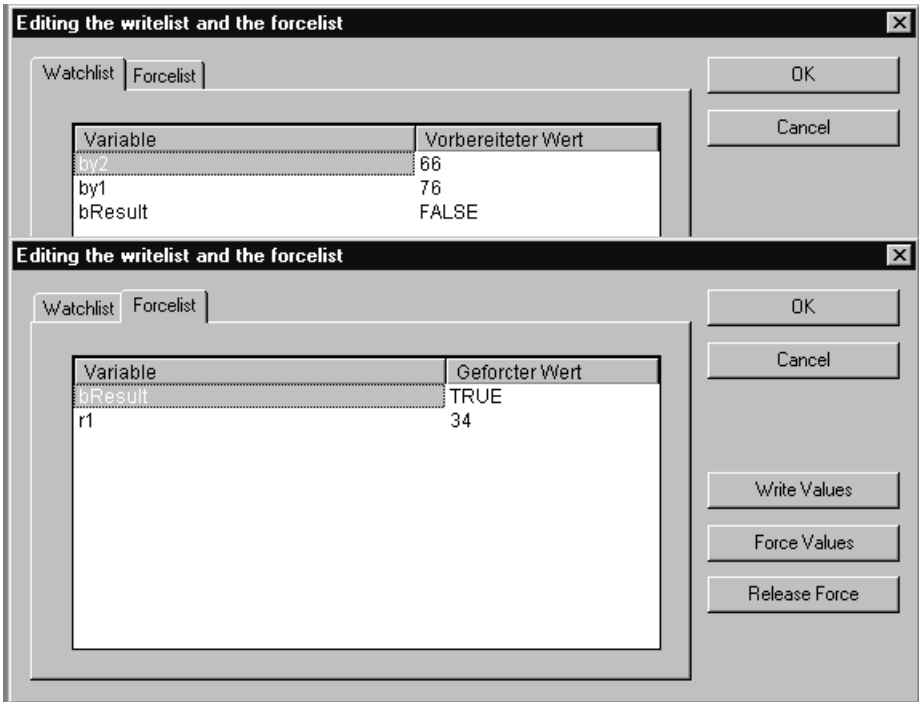
Wenn für alle gewünschten Variablen die Einstellung <Forcen aufheben> hinter dem Wert im Deklarationsfenster sichtbar ist, führen Sie den Menübefehl [Online] / [Werte forcen] ([Online] / [Force Values]) aus, der den neuen Inhalt der Forceliste auf das Programm überträgt.

4.6.18 [Online] / [Schreiben/Forcen-Dialog] ([Online] / [Write/Force-Dialog])

Tastaturbefehl: <Strg>+<Umschalt>+<F7>

Nach dem Ausführen des Menübefehls [Online] / [Schreiben/Forcen-Dialog] wird der Dialog ([Editing the writeliste and the forceliste]) geöffnet, der in zwei Registern die aktuelle Watchiste (Watchlist) und Forceliste (Forcelist) darstellt. In einer Tabelle werden jeweils der Variablename und deren zum Schreiben vorbereiteter bzw. geforcte Wert dar-

gestellt.



20761AEN

Mit dem Menübefehl [Online] / [Werte schreiben] ([Online] / [Write Values]) werden die Variablen in die Watchlist übertragen und mit dem Menübefehl [Online] / [Werte forcen] ([Online] / [Force Values)] in die Forcelist verschoben.

Die Werte der Variablen können Sie in der Forceliste in den Spalten [Vorbereiteter Wert] bzw. [Geforcter Wert] ([Prepared Value] bzw. [Forced Value]) editieren. Wenn Ihre Eingabe nicht kompatibel mit dem Variablentypen ist, erscheint eine Fehlermeldung. Wenn Sie einen Wert löschen und den Dialog verlassen (außer mit [Abbrechen] ([Cancel]), wird der Eintrag aus der Watchliste entfernt und die Variable zum Aufheben des Forcing vorgemerkt.

Folgende Menübefehle, die denen im Menü [Online] entsprechen, stehen über Schaltflächen zur Verfügung:

[Werte forcen] ([Werte forcen]): Alle Einträge der aktuelle Schreibliste werden in die Forceliste verschoben. Die Forcing-Werte der Variablen werden in die Steuerung übertragen. Für alle Variablen, die mit [Forcen aufheben] ([Release Forcing]) markiert sind, wird das Forcing nicht durchgeführt. Der Dialog wird danach geschlossen.

[Werte schreiben] ([Write Values]): Alle Einträge der aktuellen Schreibliste werden einmalig auf die entsprechenden Variablen in der Steuerung geschrieben. Der Dialog wird danach geschlossen.

[Forcen aufheben] ([Release Force]): Alle Einträge der Forceliste werden gelöscht Wenn eine Watchliste vorhanden ist, wird der Dialog [Löschen der Schreib-/Forcelisten] geöffnet. In dem Dialog können Sie festlegen, ob nur das Forcen aufgehoben oder auch die Schreibliste gelöscht werden soll. Der Dialog wird danach bzw. nach Schließen des Auswahldialogs geschlossen.

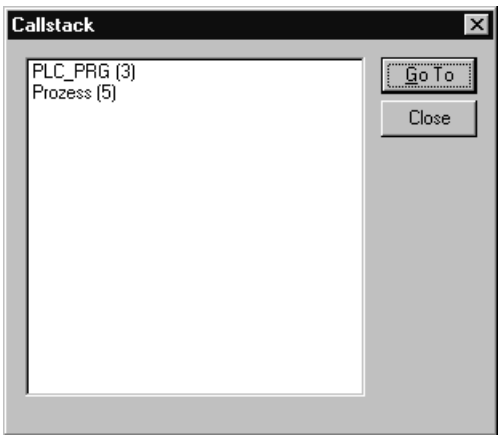
Hinweis:

Bei einigen Zielsystemen die Forceliste auf der Steuerung erhalten, auch wenn die Verbindung z. B. duch Ausloggen unterbrochen wird.

4.6.19 [Online] / [Aufrufhierarchie] ([Online] / [Show Call Stack])

Den Menübefehl [Online] / [Aufrufhierarchie] können Sie ausführen, wenn die Simulation an einem Breakpoint stoppt. Mit dem Menübefehl können Sie die Bausteine, die sich gerade im Aufruf-Stack befinden, einzeln im Editor öffnen.

Nach dem Ausführen des Menübefehls wird ein Dialog [Aufrufhierarchie] ([Aufrufhierarchie]) geöffnet.



20758AEN

Der Dialog enthält eine Liste der Bausteine, die sich momentan im Aufruf-Stack befinden.

Der letzte Baustein in der Liste ist der Baustein, der gerade abgearbeitet wird.

Markieren Sie einen Baustein und klicken Sie anschließend auf die Schaltfläche [Gehe zu] ([Go to]). Der markierte Baustein wird im Editor geöffnet und die momentan abgearbeitete Zeile bzw. das momentan abgearbeitete Netzwerk angezeigt.

4.6.20 [Online] / [Ablaufkontrolle] ([Online] / [Display Flow Control])

Mit dem Menübefehl [Online] / [Ablaufkontrolle] wird die Ablaufkontrolle aktiviert bzw. deaktiviert.

Ob die Ablaufkontrolle aktiviert bzw. deaktiviert werden kann, hängt von den Einstellungen des Zielsystems ab.

Wenn die Abkaufkontrolle aktiviert ist, wird jeder Programmschritt markiert, der während des letzten Steuerungszyklus ausgeführt wurde.

Das Zeilennummernfeld bzw. das Netzwerknummernfeld der durchlaufenen Zeilen bzw. Netzwerke wird grün dargestellt (bei Standardeinstellung).

Im AWL-Editor wird am linken Rand jeder Zeile ein zusätzliches Feld eingefügt, in dem der aktuelle Inhalt des Akkumulators angezeigt wird.

In den grafischen Editoren für FUP und KOP wird in allen Verbindungslinien, die keine booleschen Werte transportieren, ein weiteres Feld eingefügt. In diesem Feld wird der Wert angezeigt, der bei Belegung der Aus- und Eingänge über die Verbindungslinie transportiert wird. Wenn eine Verbindungslinie ausschließlich boolesche Werte transportiert, wird blau dargestellt (bei Standardeinstellung), wenn der boolesche Wert TRUE ist. So kann der Informationsfluss ständig mit verfolgt werden.

Hinweis:

Die Laufzeit eines Programmes wird durch die Ablaufkontrolle vergrößert. Dies kann bei zeitzyklischen Programmen mit hoher Auslastung zur Zyklusüberschreitung führen.

4.6.21 [Online] / [Simulation] ([Online] / [Simulation Mode])

Mit dem Menübefehl [Online] / [Simulation] starten und beenden Sie den Simulationsmodus. Wenn der Simulationsmodus läuft, erscheint ein Haken vor dem Menüpunkt.

Im Simulationsmodus läuft das Programm auf demselben PC wie der PLC-Editor. Der Simulationsmodus dient zum Testen des Projekts. Zur Kommunikation zwischen PC und Simulation wird der **Windows Message Mechanismus** benutzt.

Wenn das Programm nicht im Simulationsmodus ist, läuft das Programm auf der Steuerung. Die Kommunikation zwischen PC und Steuerung erfolgt typischerweise über die serielle Schnittstelle oder über einen Gateway.

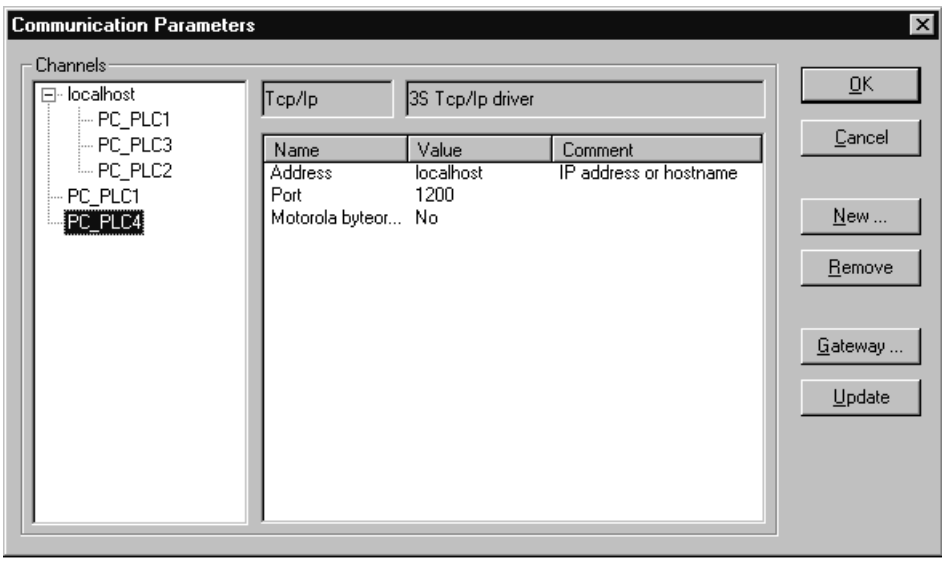
Der Status dieses Flags wird mit dem Projekt gespeichert.

Hinweis:

Bausteine aus externen Bibliotheken laufen nicht in der Simulation.

4.6.22 [Online] / [Kommunikationsparameter] ([Online] / [Communication Parameters])

Nach dem Ausführen des Menübefehls [Online] / [Kommunikationsparameter] wird der Dialog [Kommunikationsparameter] ([Communication Parameters]) geöffnet.



20759AEN

Die Einstellungen in diesem Dialog gelten für die Kommunikation zwischen dem lokalen PC und dem Laufzeitsystem über einen Gateway-Server. Bei Verwendung eines OPC- oder eines DDE-Servers müssen in deren Konfiguration dieselben Kommunikationsparameter eingestellt sein.

4.6.23 [Online] / [Quellcode laden] ([Online] / [Sourcecode download])

Mit dem Menübefehl [Online] / [Quellcode laden] laden Sie den Quellcode des Projekts in die Steuerung.

Hinweis:

Der Quellcode dürfen Sie nicht mit dem Code verwechseln, der beim Übersetzen des Projekts entsteht!

Welche Optionen für den Download gelten (Zeitpunkt, Umfang) können Sie in den Projektoptionen, Kategorie [Sourcedownload] ([Source download]) einstellen.

4.6.24 [Online] / [Bootprojekt erzeugen] ([Online] / [Create boot project])

Wenn Sie den Menübefehl [Online] / [Bootprojekt erzeugen] im Online-Modus ausführen, wird das übersetzte Projekt so auf der Steuerung abgelegt, dass die Steuerung es bei einem Neustart automatisch laden kann.

Wie die Speicherung des Boottprojekts erfolgt, ist vom Zielsystem abhängig. Beispielsweise werden auf 386er-Systemen drei Dateien angelegt:

- `default.prg` mit dem Projektcode
- `default.chk` mit der Checksumme des Projektcodes
- `default.sts` mit dem Status der Steuerung nach dem Neustart (Start/Stop).

Wenn bereits ein Bootprojekt in der Steuerung vorliegt und außerdem in den Projektoptionen, Kategorie [Arbeitsbereich] ([Desktop]), die Option [Online-Betrieb im Sicherheitsmodus] ([Online in security mode]) aktiviert ist, erscheint beim Erzeugen eines neuen Bootprojekts ein Dialog, der die Projektinformationen sowohl des aktuell im Programmiersystem geladenen als auch des auf der Steuerung liegenden Bootprojekts darstellt. Diese Funktionalität muss allerdings vom Zielsystem unterstützt werden!

Der Menübefehl [Online] / [Bootprojekt erzeugen] steht auch im Offline-Modus zur Verfügung, wenn das Projekt vorher fehlerfrei übersetzt wurde. In diesem Fall werden für das Bootprojekt eine Datei `<projektname>.prg` und für die Checksumme des Codes eine Datei `<projektname>.chk` im Projektverzeichnis angelegt. Die Daten können Sie nach Umbenennung der Dateien in `default.prg` bzw. `default.chk` auf die Steuerung laden.

In Abhängigkeit von den Zielsystemeinstellungen wird beim Erzeugen des Bootprojekts im Offline-Modus eventuell gleichzeitig eine neue `*.ri`-Datei (Download- und Übersetzungsinformationen) erzeugt. Ebenfalls vom Zielsystem abhängig ist, ob ein Nachfrage-Dialog geöffnet wird , falls bereits eine solche Datei vorliegt.

Hinweis:

Wenn in den Projektoptionen, Kategorie [Sourcedownload] ([Source download]) die Option [Implizit beim Bootprojekt erzeugen] ([Implicit on create boot project]) aktiviert ist, wird beim Menübefehl [Online] / [Bootprojekt erzeugen] der gewählte Source-Datenumfang automatisch in die Steuerung geladen.

4.6.25 [Online] / [Datei in Steuerung schreiben] ([Online] / [Write file to PLC])

Mit dem Menübefehl [Online] / [Datei in Steuerung schreiben] können Sie eine beliebige Datei in die Steuerung zu laden.

Nach dem Ausführen des Menübefehls wird der Dialog [Datei in Steuerung schreiben] ([Write file to PLC]) geöffnet, in dem Sie die gewünschte Datei markieren können. Bestätigen Sie die Auswahl mit einem Klick auf die Schaltfläche [Öffnen] ([Open]). Der Dialog wird geschlossen, die Datei in die Steuerung geladen und dort unter demselben Namen abgelegt. Das Laden wird durch eine Fortschrittsanzeige angezeigt.

Mit dem Menübefehl [Online] / [Datei aus Steuerung laden] ([Online] / [Read file from PLC]) können Sie eine auf der Steuerung abgelegte Datei auf den PC laden.

4.6.26 [Online] / [Datei aus Steuerung laden] ([Online] / [Read file from PLC])

Mit dem Menübefehl [Online] / [Datei aus Steuerung laden] können Sie eine Datei, die mit dem Menübefehl [Online] / [Datei in Steuerung schreiben] ([Online] / [Write file to PLC]) auf der Steuerung abgelegt wurde, wieder laden.

Nach dem Ausführen des Menübefehls [Online] / [Datei aus Steuerung laden] wird der Dialog [Datei aus Steuerung laden] ([Load file from PLC]) geöffnet.

Geben Sie unter [Dateiname] ([File name]) den Namen der gewünschten Datei ein und wählen Sie im Auswahlfenster das PC-Dateiverzeichnis, in das die Datei geladen werden soll.

Bestätigen Sie die Einstellungen mit der Schaltfläche [Speichern] ([Save]).

4.7 Fenster anordnen

4.7.1 Übersicht

Unter dem Menüpunkt [Fenster] ([Window]) befinden sich alle Menübefehle zur Fensterverwaltung.

Das Menübefehle der Fensterverwaltung haben folgenden Zweck:

- Automatisches Anordnen der Fenster des PLC-Editor
- Öffnen des Bibliotheksverwalters und des Logbuchs
- Wechseln zwischen den geöffneten Fenstern

Am Ende des Menüs finden Sie eine Auflistung aller geöffneten Fenster in der Reihenfolge, in der sie geöffnet wurden. Mit einem Mausklick auf einen Eintrag wechseln Sie zum jeweiligen Fenster. Vor dem Eintrag des aktiven Fensters erscheint ein Häkchen.

4.7.2 [Fenster] / [Nebeneinander] ([Window] / [Tile vertical])

Mit dem Menübefehl [Fenster] / [Nebeneinander] werden alle Fenster im Arbeitsbereich so nebeneinander angeordnet, dass sie sich nicht überlappen und den gesamten Arbeitsbereich ausfüllen.

4.7.3 [Fenster] / [Untereinander] ([Window] / [Tile horizontal])

Mit dem Menübefehl [Fenster] / [Untereinander] werden alle Fenster im Arbeitsbereich so untereinander angeordnet, dass sie sich nicht überlappen und den gesamten Arbeitsbereich ausfüllen.

4.7.4 [Fenster] / [Überlappend] ([Window] / [Cascade])

Mit dem Menübefehl [Fenster] / [Überlappend] werden alle Fenster im Arbeitsbereich kaskadenförmig übereinander angeordnet.

4.7.5 [Fenster] / [Symbole anordnen] ([Window] / [Arrange Symbols])

Mit dem Menübefehl [Fenster] / [Symbole anordnen] werden alle minimierten Fenster im Arbeitsbereich in einer Reihe am unteren Ende des Arbeitsbereichs angeordnet.

4.7.6 [Fenster] / [Alle Schließen] ([Window] / [Close All])

Mit dem Menübefehl [Fenster] / [Alle Schließen] werden alle Fenster im Arbeitsbereich geschlossen.

4.7.7 [Fenster] / [Meldungen] ([Window] / [Messages])

Tastaturbefehl: <Umschalt>+<Esc>

Mit dem Menübefehl [Fenster] / [Meldungen] wird das Meldungsfenster geöffnet oder geschlossen.

Das Meldungsfenster enthält Meldungen aus dem letzten Übersetzungs-, Überprüfungs- oder Vergleichsvorgang.

Wenn das Meldungsfenster geöffnet ist, erscheint vor dem Menübefehl ein Häkchen.

4.7.8 [Fenster] / [Bibliotheksverwaltung] ([Window] / [Library Manager])

Mit dem Menübefehl [Fenster] / [Bibliotheksverwaltung] wird der Bibliotheksverwalter geöffnet.

4.7.9 [Fenster] / [Logbuch] ([Window] / [Log])

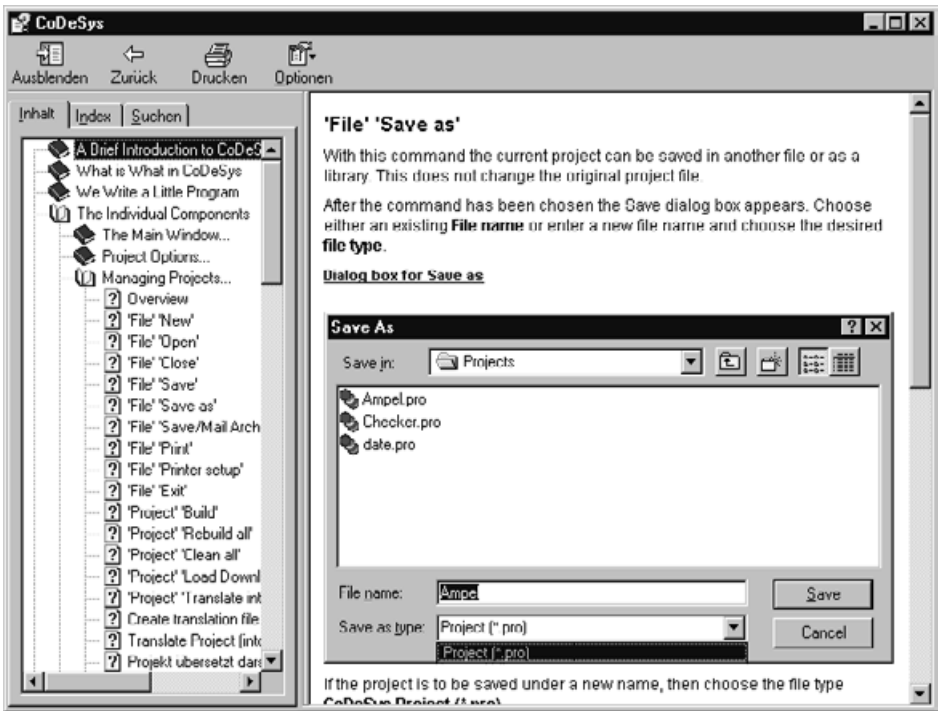
Mit dem Menübefehl [Fenster] / [Logbuch] wird das Logbuch-Fenster geöffnet.

Im Logbuch werden die Protokolle der Online-Sitzungen angezeigt.

4.8 Die Online-Hilfe

4.8.1 [Hilfe] / [Inhalt] und [Suchen] ([Help] / [Contents] und [Search])

Nach dem Ausführen der Menübefehle [Hilfe] / [Inhalt] und [Hilfe] / [Suchen] wird jeweils das Hilfethemen-Fenster geöffnet, das über den **HTML Help Viewer** (ab Internet Explorer V4.1) angezeigt wird.



hilfe_fenster

Das Fenster enthält drei Registerkarten:

- [Inhalt] ([Contents])
- [Index] ([Index])
- [Suchen] ([Search])

Die Registerkarte [Inhalt] zeigt das Inhaltsverzeichnis der Online-Hilfe. Die Bücher lassen können Sie durch Mausklick auf die Zeile mit dem Buch öffnen und schließen. Der Inhalt der im Inhaltsverzeichnis markierten Seite wird im rechten Teil des Hilfefensters angezeigt. Verknüpfungen zu anderen Hilfeseiten und aufklappbare Abschnitte sind im Hilfetext durch eine andere Farbe oder durch Unterstreichung markiert.

In der Registerkarte [Index] können Sie nach einem bestimmten Stichwort suchen.

In der Registerkarte [Suchen] können Sie eine Volltextsuche über Hilfeseiten durchführen.

Weitere Informationen finden Sie auf Seite 195 (→ Kap. 4.8.2, Kontextsensitive Hilfe).

4.8.2 Kontextsensitive Hilfe

Tastaturbefehl: <F1>

Mit der <F1>-Taste können Sie sich direkt die zum Thema passende Hilfeseite anzeigen lassen.

In folgenden Bereichen des PLC-Editors funktioniert die kontextsensitive Hilfe:

- im aktiven Fenster
- in einem Dialog
- über einem Menübefehl
- markierter Text in einem Fenster, z. B. ein Schlüsselwort, eine Standardfunktion oder eine Fehlermeldung im Meldungsfenster

5 PLC-Editor Programmierfenster

5.1 Editoren für die Programmiersprachen

5.1.1 Aufbau eines Editors

Für alle Programmiersprachen im PLC-Editor bestehen die Editoren aus einem Deklarationsteil und einem Rumpf. Der Rumpf kann aus einem Text- oder Grafikeditor bestehen. Der Deklarationsteil ist immer ein Texteditor. Der Rumpf ist durch einen horizontalen Bildschirmteiler vom Deklarationsteil getrennt. Den Bildschirmteiler können Sie nach Bedarf verschieben, indem Sie ihn mit der Maus anklicken und mit gedrückter Maustaste nach oben oder unten bewegen.

Druckgrenzen

Wenn die Option [Druckbereiche anzeigen] ([Show print area margins]) im Dialog [Projekt] / [Optionen] / [Arbeitsbereich] ([Projects] / [Options] / [Desktop]) angewählt wurde, werden die vertikalen und horizontalen Seitenbegrenzungen, die beim Drucken des Editor-Inhalts gelten, durch rot gestrichelte Linien dargestellt. Dabei gelten die Vorgaben des eingestellten Druckers, sowie die im Menü [Datei] /[Einstellungen Dokumentation] ([File] / [Printer Setup]) ausgewählte Größe der Druckvorlage. Ist kein Drucker bzw. keine Druckvorlage eingestellt, wird die Default-Belegung verwendet (Default.DFR und Standard-Drucker). Die horizontalen Druckgrenzen werden so eingezeichnet, als wäre im Dialog [Einstellungen Dokumentation] ([Printer Setup]) die Optionen [Neue Seite je Objekt] ([New page for each object]) bzw. [Neue Seite je Unterobjekt] ([New page for each subobject]) angewählt. Die unterste Grenze ist nicht dargestellt.

Beachten Sie:

Eine exakte Anzeige der Druckbereichsgrenzen ist nur gewährleistet, wenn der Zoomfaktor auf 100% eingestellt ist.

5.1.2 Kommentar

Benutzerkommentare müssen in die Zeichenfolgen "(*" und "*)" eingeschlossen werden.

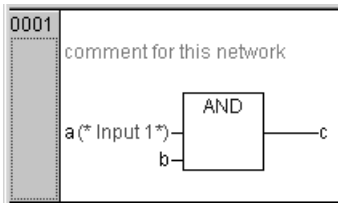
Beispiel: `(* Dies ist ein Kommentar. *)`

Kommentare sind **in allen Texteditoren** und dort an beliebiger Stelle erlaubt: in allen Deklarationen, in den Sprachen AWL und ST und in selbst definierten Datentypen. Wenn das Projekt unter Verwendung einer **Dokumentationsvorlage** ausgedruckt wird, erscheint in textbasierten Programmteilen der Kommentar, der bei der Variablendeklaration eingegeben wurde, jeweils hinter der Variable.

In den **grafischen Editoren FUP und KOP** können Sie zu jedem Netzwerk Kommentare eingeben. Wählen Sie hierzu das Netzwerk aus, das Sie kommentieren möchten und klicken Sie auf [Einfügen] / [Kommentar] ([Insert] / [Comment]). Außerdem können Sie dort, wo Sie den **Variablennamen** eingeben, Kommentare hinzufügen.

Beispiel

Netzwerk-Kommentar und Kommentar hinter einer Eingangsvariablen im FUP-Editor:



20507AXX

Im KOP-Editor können Sie zu jedem Kontakt bzw. zu jeder Spule einen Kommentar hinzufügen. Um Kommentare hinzufügen zu können, wählen Sie den Menübefehl [Extras] / [Optionen] ([Extras] / [Options]) und aktivieren Sie die Option [Kommentare pro Kontakt] ([Comments per Contact]).

Im CFC-Editor gibt es spezielle Kommentarbausteine, die Sie beliebig einfügen können.

Im AS-Editor können Sie einen Kommentar zu einem Schritt hinzufügen. Den Kommentartext geben Sie im Dialog [Extras] / [Schritte Attribute] ([Extras] / [Steps Attributes]) ein.

Wenn die entsprechende Option im Dialog [Projekt] / [Optionen] / [Übersetzungsoptionen] ([Project] / [Options] / [Build]) aktiviert ist, sind auch verschachtelte Kommentare erlaubt.

Wenn Sie den Mauszeiger im Online-Modus eine kurze Zeit über einer Variablen halten, wird der Typ, gegebenenfalls die Adresse und der Kommentar der Variablen in einem Tooltip angezeigt.

5.1.3 [Extras] / [Zoom] ([Extras] / [Zoom])

Tastaturbefehl: <Alt>+<Eingabe>

Der Menübefehl [Extras] / [Zoom] ([Extras] / [Zoom]) steht zur Verfügung, wenn der Cursor in den Texteditoren auf dem Namen eines aufgerufenen Bausteins steht bzw. wenn in den grafischen Editoren die Box eines Bausteins markiert ist. Mit diesem Menübefehl öffnen Sie den betreffenden Baustein in seinem Editorfenster.

Wenn es sich um einen Baustein aus einer Bibliothek handelt, wird der Bibliotheksverwalter aufgerufen und der entsprechende Baustein angezeigt.

5.1.4 [Extras] / [Instanz öffnen]

Der Menübefehl [Extras] / [Instanz öffnen] ([Extras] / [Open instance]) entspricht dem Menübefehl [Projekt] / [Instanz öffnen] ([Project] / [View instance]). Der Menübefehl [Extras] / [Instanz öffnen] steht zur Verfügung, wenn der Cursor in Texteditoren auf dem Namen eines Funktionsblocks steht bzw. wenn in grafischen Editoren die Box eines Funktionsblocks markiert ist.

5.1.5 Intellisense-Funktion

Wenn die Option [Komponenten auflisten] ([List components]) im Dialog [Projekt] / [Optionen] / [Editor] ([Project] / [Options] / [Editor]) aktiviert ist, steht die **Intellisense**-Funktion an folgenden Stellen zur Verfügung:

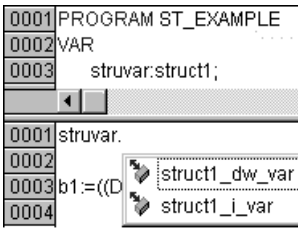
- in allen Editoren
- im Watch- und Rezepturverwalter
- in der Tracekonfiguration

Die Intellisense-Funktion unterstützt Sie bei der Programmierung durch kontextabhängige Hilfe:

- Wenn Sie anstelle eines Bezeichners einen Punkt "." eingeben, öffnet sich eine Auswahlliste aller lokalen und globalen Variablen. Aus dieser Liste können Sie ein Element auswählen und durch Drücken der <Eingabe>-Taste hinter dem Punkt einfügen. Alternativ können Sie auf das Listenelement doppelklicken.
- Wenn Sie nach einem Bezeichner für eine Funktionsblockinstanz oder eine als Struktur definierte Variable einen Punkt eingeben, öffnet sich eine Auswahlliste der Ein- und Ausgangsvariablen des Funktionsblocks bzw. der Strukturkomponenten.

Beispiel

Eingabe von "struvar." -> die Komponenten der Struktur struct1 werden angeboten:



20422AXX

- Wenn Sie eine beliebige Zeichenfolge eingeben und den Tastaturbefehl <Strg>+<Leertaste> drücken, erscheint eine Auswahlliste aller im Projekt verfügbaren Bausteine und globalen Variablen, wobei die erste, die mit dieser Zeichenfolge beginnt, markiert ist. Wenn Sie die <Eingabe>-Taste drücken, wird der markierte Baustein bzw. die Variable ins Programm übernommen.

5.2 Der Deklarationseditor

5.2.1 Allgemeines zum Deklarationseditor

Den Deklarationseditor verwenden Sie:

- zur Variablendeklaration von Bausteinen und globalen Variablen,
- zur Datentypdeklaration,
- im Watch- und Rezepturverwalter.

Wenn der entsprechende Treiber installiert ist, verfügt der Deklarationseditor über die Funktionalitäten von Windows und der IntelliMouse.

Im Überschreib-Modus wird in der Statusleiste [ÜB] ([OV]) schwarz angezeigt. Zwischen dem Überschreib- und dem Einfüge-Modus können Sie mit der <Einfg>-Taste wechseln.

Die Variablendeklaration wird durch **Syntaxcoloring** unterstützt.

Die wichtigsten Befehle finden Sie im Kontextmenü, das Sie durch einen Klick mit der rechten Maustaste öffnen.

Hinweis:

Die Eigenschaften einer Variablen während der Übersetzung bzw. dem Pre-Compile können mit Hilfe von Pragmas beeinflusst werden.

Weitere Informationen finden Sie in den folgenden Kapiteln:

- (→ Kap. 5.2.2, Deklarationsteil), Seite 200
- (→ Kap. 5.2.3, Eingabevariablen), Seite 201
- (→ Kap. 5.2.4, Ausgabevariablen), Seite 201
- (→ Kap. 5.2.5, Ein-/Ausgabevariablen), Seite 201
- (→ Kap. 5.2.6, Lokale Variablen), Seite 202
- (→ Kap. 5.2.7, Remanente Variablen), Seite 202
- (→ Kap. 5.2.8, Konstanten (Typed Literals)), Seite 203
- (→ Kap. 5.2.9, Externe Variablen), Seite 204
- (→ Kap. 5.2.10, Schlüsselwörter), Seite 204
- (→ Kap. 5.2.11, Variablendeklaration), Seite 204
- (→ Kap. 5.2.12, AT-Deklaration), Seite 205
- (→ Kap. 5.2.13, [Einfügen] / [Deklarationsschlüsselworte]), Seite 205
- (→ Kap. 5.2.14, [Einfügen] / [Typen]), Seite 205
- (→ Kap. 5.2.15, Syntaxcoloring), Seite 206
- (→ Kap. 5.2.16, Kurzform-Modus), Seite 206
- (→ Kap. 5.2.17, Automatisch deklarieren), Seite 207
- (→ Kap. 5.2.18, Zeilennummer im Deklarationseditor), Seite 209
- (→ Kap. 5.2.19, Deklaration als Tabelle), Seite 209
- (→ Kap. 5.2.20, [Einfügen] / [Neue Deklaration] ([Insert] / [New Declaration])), Seite 210
- (→ Kap. 5.2.21, Deklaration im Online-Modus), Seite 210

5.2.2 Deklarationsteil

Im Deklarationsteil eines Bausteins werden alle Variablen deklariert, die nur in diesem Baustein verwendet werden. Dies können folgende Variablen sein:

- Eingabevariablen
- Ausgabevariablen
- Ein-/Ausgabevariablen
- lokale Variablen
- remanente Variablen
- Konstanten

Die Deklarationssyntax orientiert sich an der Norm IEC 61131-3.

Bei folgenden Objekttypen können Sie Objektvorlagen für das initiale Ausfüllen des Deklarationsteils verwenden:

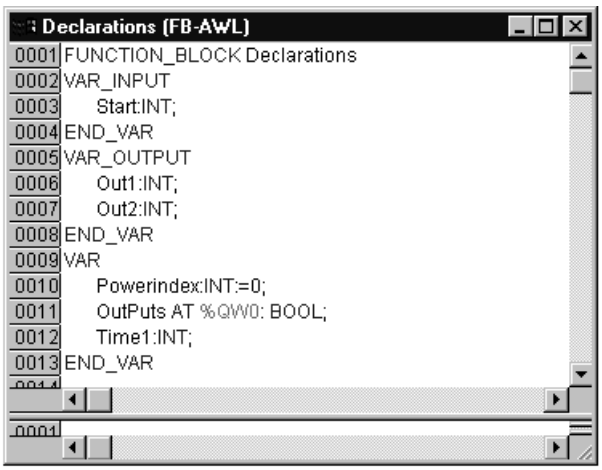
- Dateityp
- Funktion
- Funktionsbaustein
- Globale Variablen
- Programm

Hinweis:

Die Eigenschaften einer Variablen während der Übersetzung bzw. dem Pre-Compile können sie mit Hilfe von Pragmas beeinflussen.

Beispiel

Korrekte Variablendeklaration im PLC-Editor.



20423AEN

5.2.3 Eingabevariablen

Zwischen den Schlüsselwörtern VAR_INPUT und END_VAR werden alle Variablen deklariert, die als Eingabevariablen eines Bausteins dienen. An der Aufrufstelle kann der Wert der Variablen beim Aufruf mitgegeben werden.

Beispiel

```
VAR_INPUT
   in1:INT; (* 1. Eingabevariable *)
END_VAR
```

Beispiel: Zugriff auf eine Eingabevariable eines Funktionsblocks

Der Funktionsblock FUB hat eine Eingabevariable in1 vom Typ int.

Deklaration

```
PROGRAM prog
   VAR
      inst:FUB;
   END_VAR
```

Programmteil in AWL

```
LD  17
ST  inst.in1
CAL inst
```

Programmteil in ST

```
inst(in1:=17);
```

5.2.4 Ausgabevariablen

Zwischen den Schlüsselwörtern VAR_OUTPUT und END_VAR werden alle Variablen deklariert, die als Ausgabevariablen eines Bausteins dienen. Die Werte dieser Variablen werden an den aufrufenden Baustein zurückgeliefert. Der aufrufende Baustein kann die Werte der Variablen abgefragen und weiterverwenden.

Beispiel

```
VAR_OUTPUT
   out1:INT; (* 1. Ausgabevariable *)
END_VAR
```

5.2.5 Ein-/Ausgabevariablen

Zwischen den Schlüsselwörtern VAR_IN_OUT und END_VAR werden alle Variablen deklariert, die als Ein- und Ausgabevariablen eines Bausteins dienen.

Achtung:

Bei diesen Variablen wird der Wert der übergebenen Variablen direkt verändert (Übergabe als Pointer, Call-by-Reference). Deshalb darf der Eingabewert für eine solche Variable keine Konstante sein!

VAR_IN_OUT-Variablen eines Funktionsblocks können nicht von außen über <Funktionsblockinstanz>.<Ein-/Ausgabevariable> gelesen oder beschrieben werden!

Beispiel

```
VAR_IN_OUT
   inout1:INT; (* 1. Ein/Ausgabevariable *)
END_VAR
```

5.2.6 Lokale Variablen

Zwischen den Schlüsselwörtern VAR und END_VAR werden alle lokalen Variablen eines Bausteins deklariert. Lokale Variablen haben keine Verbindung nach außen. Von außen kann auf diese Variablen nicht geschrieben werden.

Beispiel

```
VAR
   loc1:INT; (* 1. lokale Variable *)
END_VAR
```

5.2.7 Remanente Variablen

Remanente Variablen können ihren Wert über die übliche Programmlaufzeit hinaus beibehalten. Dazu gehören Retain-Variablen und Persistente Variablen.

Beispiel

```
VAR RETAIN
   rem1:INT; (* 1. Remanente Variable *)
END_VAR
```

Retain-Variablen

Retain-Variablen werden mit dem Schlüsselwort RETAIN gekennzeichnet. Diese Variablen behalten ihren Wert nach dem unkontrolliertem Beenden wie auch nach normalem Aus- und Einschalten der Steuerung (entsprechend dem Menübefehl [Online] / [Reset] ([Online] / [Reset]). Bei erneutem Start des Programms wird mit den gespeicherten Werten weitergearbeitet. Ein Anwendungsbeispiel ist ein Stückzähler in einer Fertigungs-Anlage, der nach einem Stromausfall weiterzählen soll.

Alle anderen Variablen werden in diesem Fall neu initialisiert, entweder mit ihren initialisierten Werten oder mit den Standard-Initialisierungen.

Retain-Variablen werden bei Ausführen der Menübefehle [Online] / [Reset Kalt] ([Online] / [Reset cold]) bzw. [Online] / [Reset Ursprung] ([Online] / [Reset original]) und bei einem erneuten Programm-Download neu initialisiert.

Persistente Variablen

Persistente Variablen werden mit dem Schlüsselwort PERSISTENT gekennzeichnet. Sie behalten ihren Wert nur nach einem erneuten Download mit dem Menübefehl [Online] / [Laden] ([Online] / [Download]). Im Gegensatz zu Retain-Variablen behalten sie ihren Wert jedoch nicht bei, wenn folgende Menübefehle ausgeführt werden:

- [Online] / [Reset] ([Online] / [Reset])
- [Online] / [Reset Ursprung] ([Online] / [Reset cold])
- [Online] / [Reset Kalt] ([Online] / [Reset original])

Persistente Variablen werden nicht im Retain-Bereich gespeichert. Sollen persistente Variablen nach einem unkontrollierten Steuerungsausfall ihre vorherigen Werte behalten, müssen persistente Variablen zusätzlich als VAR RETAIN deklariert werden. Ein Anwendungsbeispiel für persistente Retain-Variablen ist ein Betriebsstundenzähler, der nach einem Stromausfall weiterzählen soll.

nach Befehl [Online]		Var	VAR RETAIN	VAR PERSISTENT	VAR RETAIN PERSISTENT VAR PERSISTENT RETAIN
[Reset]	[Reset]	-	x	-	x
[Reset Kalt]	[Reset cold]	-	-	-	-
[Reset Ursprung]	[Reset Original]	-	-	-	-
[Laden]	[Download]	-	-	x	x
[Online Change]	[Online Change]	x	x	x	x

x = Wert bleibt erhalten - = Wert wird neu initialisiert

Achtung:

- Wenn eine lokale Variable in einem Programm als RETAIN deklariert ist, wird genau diese Variable im Retain-Bereich gespeichert (wie eine globale Retain-Variable).
- Wenn eine lokale Variable in einem Funktionsblock als RETAIN deklariert ist, wird die komplette Instanz dieses Funktionsblocks im Retain-Bereich gespeichert (alle Daten des Bausteins). Allerdings wird nur die deklarierte Retain-Variable als Retain-Variable behandelt.
- Wenn eine lokale Variable in einer Funktion als RETAIN deklariert ist, hat dies keine Auswirkung. Die Variable wird nicht im Retain-Bereich gespeichert!
- Wenn eine lokale Variable in einer Funktion als PERSISTENT deklariert wird, bleibt dies ebenfalls ohne Wirkung!

5.2.8 Konstanten (Typed Literals)

Konstanten werden mit dem Schlüsselwort CONSTANT gekennzeichnet. Konstanten können lokal oder global deklariert werden.

Syntax

```
VAR CONSTANT bzw. VAR_Global CONSTANT
   <Bezeichner>: <Typ> := <Initialisierung>
END_VAR
```

Beispiel

```
VAR CONSTANT
   con1: INT := 12; (* 1. Konstante *)
END_VAR
```

Eine Auflistung möglicher Konstanten und Information zur Verwendung von typisierten Konstanten (Typed Literals) finden Sie auf Seite 378 (→ Kap. 8, Operanden), bzw. unter dem Indexstichwort "Konstanten".

Hinweis:

Die Bezeichner von Konstanten dürfen nicht mit den nachfolgend aufgeführten Zeichenfolgen beginnen:

- MDX_
- MX_
- MC07_
- MM_

5.2.9 Externe Variablen

Globale Variablen, die in einen Baustein importiert werden sollen, werden mit dem Schlüsselwort EXTERNAL gekennzeichnet. Sie erscheinen online auch im Watchfenster des Deklarationseditors.

- Wenn die Deklaration unter VAR_EXTERNAL nicht mit der globalen Deklaration übereinstimmt, erscheint beim Übersetzen die Fehlermeldung: "Deklaration von <Name> stimmt nicht mit globaler Deklaration überein!".
- Wenn die globale Variable nicht existiert, erscheint die Meldung: "Unbekannte globale Variable: <variable>!".

Beispiel

```
VAR_EXTERNAL
   varext1: INT := 12; (* 1st external value *)
END_VAR
```

5.2.10 Schlüsselwörter

Schlüsselwörter dürfen Sie in allen Editoren nur in Großbuchstaben schreiben. Schlüsselwörter dürfen Sie nicht als Variablennamen verwenden.

5.2.11 Variablendeklaration

Syntax

```
<Bezeichner> {AT <Adresse>} :<Typ> {:= <Initialisierung>};
```

Die Teile in geschweiften Klammern {} sind optional.

Der **Bezeichner** ist der Name der Variablen.

Folgendes müssen Sie bei der Vergabe von Bezeichnern berücksichtigen:

- Bezeichner dürfen keine Leerstellen und Umlaute enthalten.
- Bezeichner dürfen nicht doppelt deklariert werden.
- Bezeichner dürfen nicht identisch mit einem Schlüsselwort sein.

Die Groß-/Kleinschreibung bei Variablen wird nicht beachtet. *VAR1*, *Var1* und *var1* sind also keine unterschiedlichen Variablen.

Unterstriche sind in Bezeichnern signifikant, z. B. werden *A_BCD* und *AB_CD* als unterschiedliche Bezeichner interpretiert. Mehrfach aufeinander folgende Unterstriche am Anfang eines Bezeichners oder in einem Bezeichner sind nicht erlaubt.

Die Länge des Bezeichners und der signifikante Bereich sind unbegrenzt.

Alle Variablendeklarationen und Datentypelemente können Initialisierungen (Zuweisung eines initialen Werts) enthalten. Sie erfolgen mit dem Zuweisungsoperator :=. Für Variablen von elementaren Typen sind diese Initialisierungen Konstanten. Die Default-Initialisierung ist für alle Deklarationen gleich 0.

Beispiel

```
var1: INT := 12; (* Integer-Variable mit Initialwert 12 *)
```

Wenn Sie eine Variable direkt an eine bestimmte Adresse binden möchten, dann müssen Sie die Variable mit dem Schlüsselwort AT deklarieren.

Zur schnelleren Eingabe von Deklarationen verwenden Sie den Kurzform-Modus.

In Funktionsblöcken können Sie Variablen auch mit unvollständigen Adressangaben spezifizieren. Um solche Variablen in einer lokalen Instanz nutzen zu können, müssen Sie einen Eintrag in der Variablenkonfiguration (Ressourcen) vorgenehmen.

Beachten Sie, dass Sie Variablen auch **automatisch deklarieren** können.

Hinweis:

Beachten Sie, dass Sie die Eigenschaften einer Variable für den Übersetzungslauf bzw. den Pre-Compile auch mit Hilfe von Pragmas beeinflussen können.

5.2.12 AT-Deklaration

Wenn Sie eine Variable direkt an eine bestimmte Adresse binden wollen, dann müssen Sie die Variable mit dem Schlüsselwort AT deklarieren. Der Vorteil dieser Vorgehensweise ist, dass Sie einer Adresse einen aussagekräftigeren Namen geben können, und dass Sie die Änderung eines Ein- oder Ausgangssignals nur an einer Stelle (nämlich in der Deklaration) ändern müssen.

Beachten Sie, dass Sie Variablen, die auf einen Eingang gelegt sind, nicht überschreiben können.

Beispiel

```
schalter_heizung7 AT %QX0.0: BOOL;
lichtschrankenimpuls AT %IW2: WORD;
ablage AT %MX2.2: BOOL;
```

Hinweis:

Wenn boolesche Variablen an eine BYTE-, WORD- oder DWORD-Adresse übertragen werden, belegen sie 1 Byte mit TRUE bzw. FALSE und nicht nur das erste Bit nach dem Offset!

5.2.13 [Einfügen] / [Deklarationsschlüsselworte]

Mit dem Menübefehl [Einfügen] / [Deklarations Schlüsselworte] ([Insert] / [Deklaration Keywords]) öffnen Sie eine Liste aller Schlüsselwörter, die Sie im Deklarationsteil eines Bausteins benutzen können. Nachdem Sie ein Schlüsselwort ausgewählt und die Wahl bestätigt haben, wird das ausgewählte Schlüsselwort an der aktuellen Cursorposition eingefügt.

Die Liste aller Schlüsselwörter erhalten Sie auch, wenn Sie die Eingabehilfe mit der <F2>-Taste öffnen und die Kategorie [Deklarationen] auswählen.

Die Liste aller im PLC-Editor gültigen Schlüsselworte finden Sie auf Seite 430 (→ Kap. 11, Anhang).

5.2.14 [Einfügen] / [Typen]

Mit dem Menübefehl [Einfügen] / [Typen] ([Insert] / [Types]) öffnen Sie eine Auswahl aller möglichen Typen zur Variablendeklaration.

Diese Auswahl erhalten Sie auch über die Eingabehilfe, die Sie mit der <F2>-Taste öffnen.

Die Typen sind in folgende Kategorien eingeteilt:

- Standard-Typen, z. B. BOOL, BYTE
- Definierte Typen, z. B. Strukturen, Aufzählungstypen
- Standard-Funktionsblöcke für Instanzdeklarationen
- Definierte Funktionsblöcke für Instanzdeklarationen

Der PLC-Editor unterstützt alle Standard-Typen gemäß der Norm IEC 61131-3.

Beispiele für die Verwendung der verschiedenen Typen finden Sie auf Seite 389 (→ Kap. 9, Datentypen).

5.2.15 Syntaxcoloring

Beim Syntaxcoloring sind in allen Editoren die Texte bei der Implementierung und der Variablendeklaration farbig dargestellt. Syntaxcoloring hilft Ihnen, Fehler zu vermeiden bzw. schneller zu entdecken.

Beispiele

Einen ungeschlossenen Kommentar, der dadurch Anweisungen auskommentiert, bemerken Sie sofort.

Schlüsselwörter werden nicht versehentlich falsch geschrieben.

Die Farbgebung des Syntaxcoloring hat folgende Bedeutung:

- **Blau** Schlüsselwörter
- **Grün** Kommentare
- **Rosa** Spezielle Konstanten (z.B. TRUE/FALSE, T#3s, %IX0.0)
- **Rot** Fehlerhafte Eingabe
  (z. B. ungültige Zeitkonstante, Schlüsselwort kleingeschrieben)
- **Schwarz** Variablen, Konstanten, Zuweisungsoperatoren usw.

5.2.16 Kurzform-Modus

Der Deklarationseditor des PLC-Editors verfügt über einen Kurzform-Modus.

Den Kurzform-Modus aktivieren Sie, indem Sie am Ende einer Zeile den Tastaturbefehl <Strg>+<Eingabe> ausführen.

Der Deklarationseditor des PLC-Editors unterstützt folgende Kurzformen:

- Alle Bezeichner mit Ausnahme des letzten Bezeichners einer Zeile werden zu Variablenbezeichnern der Deklaration erklärt.
- Der Typ der Deklaration wird vom letzten Bezeichner der Zeile bestimmt, hierbei gilt:
  - B oder BOOL ergibt BOOL
  - I oder INT ergibt INT
  - R oder REAL ergibt REAL
  - S oder STRING ergibt STRING
- Wenn anhand dieser Regeln kein Typ festgelegt werden konnte, dann wird der Typ der Variablen auf BOOL festgelegt und der letzte Identifikator wird nicht als Typ benutzt (Beispiel 1 in der nachfolgenden Tabelle).
- Jede Konstante wird, je nach Typ der Deklaration, zu einer Initialisierung oder einer Stringlänge konvertiert (Beispiele 2 und 3 in der nachfolgenden Tabelle).

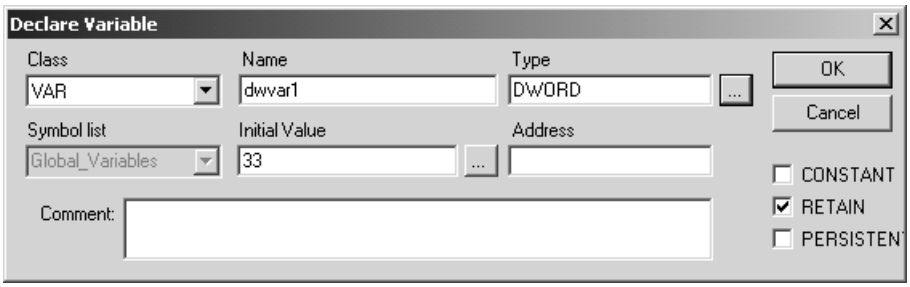
- Eine Adresse (z. B. %MD12) wird um das AT-Attribut erweitert (Beispiel 4).
- Ein Text nach einem Strichpunkt ";" wird in einen Kommentar konvertiert (Beispiel 4).
- Alle anderen Zeichen in der Zeile werden ignoriert (z. B. das Ausrufezeichen in Beispiel 5).

Beispiele

Nr.	Kurzform	Deklaration
1	`A`	`A: BOOL;`
2	`A B I 2`	`A, B: INT := 2;`
3	`ST S 2; Ein String`	`ST: STRING(2); (* Ein String *)`
4	`X %MD12 R 5;  Reelle Zahl`	`X AT %MD12: REAL := 5.0; (* Reelle Zahl *)`
5	`B !`	`B: BOOL;`

5.2.17 Automatisch deklarieren

Wenn in den Projektoptionen, Kategorie [Editor] ([Editor]) die Option [Automatisch deklarieren] ([Autodeclaration]) aktiviert ist, erscheint in allen Editoren nach Eingabe einer noch nicht deklarierten Variablen ein Dialog, mit dessen Hilfe die Variable deklarieren können.



20400AEN

Wählen Sie im Auswahlfeld [Klasse] ([Class]) einen der folgenden Variablentypen aus:

- lokale Variable (VAR)
- Eingabevariable (VAR_INPUT)
- Ausgabevariable (VAR_OUTPUT)
- EinAusgabevariable (VAR_IN_OUT)
- globale Variable (VAR_GLOBAL)

Legen Sie anschließend mit einer der folgenden Optionen fest ob es sich um eine Konstante oder eine remanente Variable handelt:

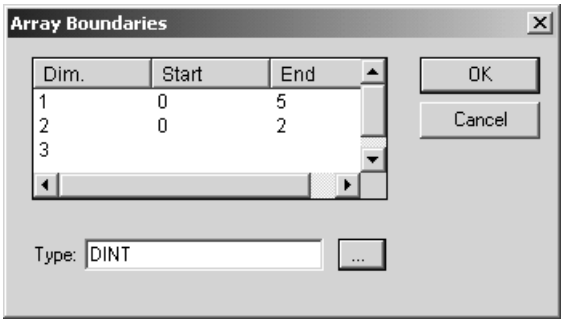
- [CONSTANT],
- [RETAIN],
- [PERSISTENT],

Das Feld [Name] ([Name]) ist mit dem im Editor eingegebenen Variablennamen vorbelegt.

Das Feld [Typ] ([Type]) ist zunächst mit BOOL belegt. Wenn Sie auf die Schaltfläche [...] klicken, wird der Dialog [Eingabehilfe] ([Help Manager]) geöffnet. In dem Dialog sind alle Datentypen aufgelistet, die für die Variable erlaubt sind. Wählen Sie den Datentyp aus und bestätigen Sie mit [OK].

Deklaration von Arrays (Feldern)

Wenn Sie im Feld [Typ] ([Type]) den Typ ARRAY ausgewählt haben, wird der Dialog [Array-Grenzen] ([Array Boundaries]) geöffnet.



20401AEN

Geben Sie für jede Dimension [Dim.] in den Feldern [Start] ([Start]) und [Ende] ([End]) die Grenzen ein.

Geben Sie im Feld [Typ] ([Type]) den Datentyp des Arrays ein. Wenn Sie auf die Schaltfläche [...] klicken, wird der Dialog [Eingabehilfe] ([Help Manager]) geöffnet, in dem die erlaubten Datentypen aufgelistet sind.

Bestätigen Sie den Dialog [Array-Grenzen] ([Array Boundaries]) mit [OK]. Mit den eingegebenen Daten wird das Feld [Typ] ([Type]) im Dialog [Variablendeklaration] ([Declare Variable]) im IEC-Format belegt.

Beispiel

```
ARRAY [1..5, 1..3] OF INT
```

Geben Sie anschließend im Feld [Initialwert] ([Initial Value]) den Initialwert der zu deklarierenden Variable ein.

Dazu benutzen Sie am besten den Dialog [Eingabehilfe] ([Help Manager]), den Sie über die Schaltfläche [...]> oder die <F2>-Taste öffnen.

- Wenn die zu deklarierenden Variable vom Typ ARRAY ist, enthält der Dialog eine Liste der Array-Elemente. Klicken Sie auf die Stelle hinter ":=". und geben Sie den Initialwert ein.
- Wenn die zu deklarierende Variable eine Struktur ist, werden die einzelnen Komponenten in Baumstruktur dargestellt. Hinter dem Variablennamen stehen in Klammern Typ und Default-Initialwert der Komponente. dahinter folgt jeweils ":=". Klicken Sie auf die Stelle hinter ":=" und geben Sie den gewünschten Initialwert ein. Wenn eine Komponente als ARRAY deklariert ist, klicken Sie auf das Pluszeichen vor dem Array-Namen und geben Sie für die Array-Elemente den Initialwert ein.

Bestätigen Sie den Dialog [Eingabehilfe] mit der [OK]. Im Dialog [Variablendeklaration] ([Declare Variable]) erscheint anschließend im Feld [Initialwert] ([Inital Value]) die Initialwerte im IEC-Format.

Beispiel

```
x:=5,feld:=2,3,struct2:=(a:=2,b:=3)
```

Im Feld [Adresse] ([Address]) können Sie die zu deklarierende Variable an eine IEC-Adresse binden (AT-Deklaration).

Geben Sie bei Bedarf einen Kommentar im Feld [Kommentar] ([Comment]) ein. Innerhalb dieses Felds einfügen Sie Zeilenumbrüche mit dem Tastaturbefehl <Strg>+<Eingabe> ein.

Schließen Sie den Dialog [Variablendeklaration] ([Declare Variable]) mit [OK]. Die Variable wird gemäß der IEC-Syntax in den entsprechenden Deklarationseditor eingetragen.

Hinweis:

Den Dialog [Variablendeklaration] ([Declare Variable]) können Sie auch mit dem Menüefehl [Bearbeiten] / [Variablen Deklaration] ([Edit] / [Auto Declare]) öffnen. Informationen dazu finden Sie auf Seite 167 (→ Kap. 4.5, Allgemeine Editierfunktionen).

Wenn der Cursor auf einer Variablen steht, können Sie diesen Dialog mit den aktuellen variablenbezogenen Einstellungen mit dem Tastaturbefehl <Umschalt>+<F2> öffnen.

5.2.18 Zeilennummer im Deklarationseditor

Im **Offline-Modus** markieren Sie durch einen einfachen Klick auf eine spezielle Zeilennummer die gesamte Textzeile.

Im **Online-Modus** öffnen Sie durch einen einfachen Klick auf eine bestimmte Zeilennummer die Eingabehilfe der Variable. Dies gilt nur, wenn es sich um eine strukturierte Variable handelt.

5.2.19 Deklaration als Tabelle

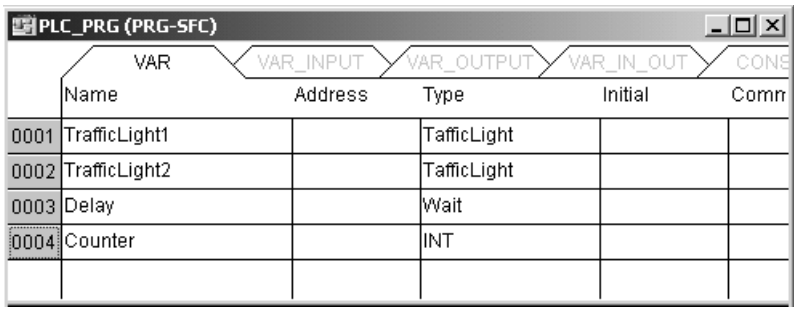
Wenn in den Projektoptionen, Kategorie [Editor] ([Editor]) die Option [Deklarationen als Tabelle] ([Declarations as tables]) aktiviert ist, wird der Deklarationseditor tabellarisch dargestellt. Sie können die einzelnen Registerkarten der jeweiligen Variablenarten wie in einem Karteikasten auswählen und die Variablen eingeben.

Zur Eingabe stehen Ihnen für jede Variable folgende Felder zur Verfügung:

[Name]	[Name]	Geben Sie den Bezeichner der Variablen ein.
[Adresse]	[Address]	Geben Sie gegebenenfalls die Adresse der Variablen ein (AT-Deklaration).
[Typ]	[Type]	Geben Sie den Typ der Variablen ein. Bei der Deklaration eines Funktionsblocks geben Sie den Funktionsblock ein.
[Initial]	**[Initial]**	Bei Bedarf geben Sie die Initialisierung der Variablen ein (entsprechend dem Zuweisungsoperator " := ").
[Kommentar]	[Comment]	Bei Bedarf geben Sie einen Kommentar ein.

Zwischen den beiden Darstellungsarten des Deklarations-Editors können Sie problemlos wechseln. Im Online-Modus gibt es für die Darstellung des Deklarations-Editors keine Unterschiede.

Um eine neue Variable einzugeben, führen Sie den Menübefehl [Einfügen] / [Neue Deklaration] ([Insert] / [New Declaration]) aus.



20402AEN

5.2.20 [Einfügen] / [Neue Deklaration] ([Insert] / [New Declaration])

Mit dem Menübefehl [Einfügen] / [Neue Deklaration] fügen Sie eine neue Variable in die Deklarationstabelle des Deklarationseditors ein. Wenn der Cursor sich in einem Tabellenfeld befindet, wird die neue Variable vor dieser Zeile eingefügt, ansonsten an das Ende der Tabelle.

Sie können an das Ende der Tabelle eine neue Deklaration anfügen, indem Sie im letzten Feld der Tabelle die rechte Pfeiltaste oder die <Tabulator>-Taste drücken.

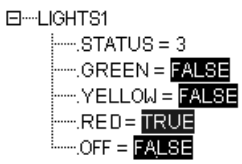
Das Feld [Name] ([Name]) ist zunächst mit "Name" und das Feld [Typ] ([Type]) mit "Bool" belegt. Ändern Sie diese Werte entsprechend. Die Eingabe des Namens und des Typs sind für die vollständige Variablendeklaration ausreichend.

5.2.21 Deklaration im Online-Modus

Im Online-Modus wird der Deklarationseditor in einem Monitor-Fenster dargestellt. In jeder Zeile steht eine Variable, gefolgt von einem Gleichheitszeichen "=" und dem Wert der Variablen. Wenn die Variable zu diesem Zeitpunkt undefiniert ist, erscheinen drei Fragezeichen "???". Bei Funktionsblöcken werden die Werte nur für geöffnete Instanzen (→ Kap. 4.4.12, [Projekt] / [Instanz öffnen] ([Project] / [View Instance])) angezeigt.

Vor jeder Variablen mit mehreren Elementen steht ein Pluszeichen. Um den Inhalt der Variablen sichtbar zu machen, klicken Sie auf das Pluszeichen, drücken Sie die <Eingabe>-Taste oder doppelklicken Sie auf die Variable.

Beispiel für aufgeklappte Variable



20424AXX

Bei einer aufgeklappten Variablen werden alle Komponenten der Variablen aufgelistet.

Um die Variable wieder zu schließen, klicken Sie auf das Minuszeichen vor der Variablen..

Um den Dialog zur Eingabe einer Variablen zu öffnen, drücken Sie die <Eingabe>-Taste drücken oder doppelklicken Soe auf eine Variable mit einem Element. In diesem Dialog können Sie den aktuellen Wert der Variablen ändern.

Bei booleschen Variablen erscheint kein Dialog. Den Wert von booleschen Variablen ändern Sie durch Doppelklicken.

Wenn Sie auf die boolesche Variable doppelklicken, wird der neue Wert hinter der Variable in türkisfarbener Schrift und spitzen Klammern angezeigt und bleibt unverändert.	bvar = TRUE < := FALSE> ivar = 509 < := 65> 20509AXX
Mit dem Menübefehl [Online] / [Werte schreiben] ([Online] / [Write Values]) setzen Sie alle Variablen auf die gewählten Werte. Die Variablen werden dann wieder schwarz dargestellt.	bvar = TRUE ivar = 509 20510AXX
Mit dem Menübefehl [Online] / [Werte forcen] ([Online] / [Force Values]) setzen Sie alle Variablen auf die gewählten Werte. Die Farbe des Force-Werts wechselt auf rot. Mit dem Menübefehl [Online] / [Forcen aufheben] ([Online] / [Release Force]) setzen Sie alle Variablen auf ihre vorherigen Werte.	bvar = FALSE ivar = 44 20511AXX

Mit dem Menübefehl [Online] / [Forcen aufheben] ([Online] / [Release Force]) setzen Sie alle Variablen auf ihre vorherigen Werte.	

5.3 Pragma-Anweisungen

5.3.1 Pragmas, Allgemeines

Eine Pragma-Anweisung dient zum Steuern des Übersetzungsvorgangs. Sie steht mit zusätzlichem Text in einer Programmzeile oder in einer eigenen Zeile des Deklarationseditors.

Eine Pragma-Anweisung wird in geschweifte Klammern gesetzt. Groß- oder Kleinschreibung wird nicht berücksichtigt.

Syntax

```
{ <Anweisungstext> }
```

Wenn der Compiler den Anweisungstext nicht sinnvoll interpretieren kann, wird das gesamte Pragma wie ein Kommentar behandelt und überlesen. Der Compiler gibt jedoch eine Warnung aus.

Abhängig vom Typ und Inhalt wirkt ein Pragma auf die Zeile, in der es steht, oder auf alle folgenden Zeilen.

Ein Pragma ist wirksam, bis eines der folgenden Ereignisse eintritt:

- Das Pragma wird durch ein neues Pragma aufgehoben.
- Dasselbe Pragma wird mit anderen Parametern ausgeführt.
- Das Ende eines Blocks wird erreicht.

Als Block ist hierbei Folgendes gemeint:

- Deklarationsteil
- Implementationsteil
- Globale Variablenliste
- Typdeklaration

Die öffnende Klammer darf unmittelbar auf einen Variablennamen folgen. Die öffnende und die schließende Klammer müssen sich in derselben Zeile befinden.

Folgende Pragmas können Sie im PLC-Editor verwenden:

- Pragmas zur Initialierung, Monitoring, Symbolerzeugung auf Seite 212
- Pragmas für den Bit-Zugriff auf Seite 214
- Pragmas zur Anzeige/Nicht-Anzeige von Deklarationsteilen im Bibliotheksverwalter auf Seite 215

5.3.2 Pragmas für Initialisierung, Monitoring, Symbolerzeugung und für den Bit-Zugriff

Allgemeine Informationen zu Pragmas finden Sie auf Seite 212 (→ Kap. 5.3.1, Pragmas, Allgemeines).

Pragma {flag}

Das Pragma {flag} wird für Initialisierung, Monitoring und Symbolerzeugung verwendet.

Syntax

```
{flag [<flags>] [off/on]}
```

Mit diesem Pragma können Sie die Eigenschaften einer Variablendeklaration beeinflussen.

<flags> kann eine Kombination der folgenden Flags sein:

noinit	Die Variable wird nicht initialisiert.
nowatch	Die Variable wird nicht überwacht (kein Monitoring).
noread	Die Variable wird ohne Leserecht in die Symboldatei exportiert.
nowrite	Die Variable wird ohne Schreibrecht in die Symboldatei exportiert.
noread, nowrite	Die Variable wird nicht in die Symboldatei exportiert.

Mit der Modifikation on wirkt das Pragma auf alle folgenden Variablendeklarationen, bis es vom Pragma {flag off} aufgehoben wird, bzw. bis es von einem anderen Pragma {flag <flags> on} überschrieben wird.

Ohne die Modifikation on oder off wirkt das Pragma nur auf die aktuelle Variablendeklaration, die mit dem nächsten Strichpunkt abgeschlossen wird.

Beispiele für die Verwendung des Pragmas {flag}

Initialisierung und Überwachung (Monitoring) von Variablen

Beispiel 1

Die Variable a wird nicht initialisiert und nicht überwacht (monitored).

Die Variable b wird nicht initialisiert.

```
VAR
   a : INT {flag noinit, nowatch};
   b : INT {flag noinit };
END_VAR
VAR
   {flag noinit, nowatch on}
   a : INT;
   {flag noinit on}
   b : INT;
   {flag off}
END_VAR
```

Beispiel 2

Beide Variablen werden nicht initialisiert.

```
{flag noinit on}
VAR
   a : INT;
   b : INT;
END_VAR
{flag off}
VAR
   {flag noinit on}
   a : INT;
   b : INT;
   {flag off}
END_VAR
```

Variablenexport in die Symboldatei

Die Flags noread und nowrite dienen dazu, in einem Baustein ohne Lese- und/oder Schreibrecht einzelne Variablen mit einem eingeschränkten Zugriffsrecht auszustatten. Die Defaulteinstellung einer Variable ist die Einstellung des Bausteins, in dem die Variable deklariert wurde. Wenn eine Variable weder Lese- noch Schreibrechte hat, wird sie nicht in die Symboldatei exportiert.

Beispiel 1

Der Baustein wird mit Lese- und Schreibrecht ausgestattet. Mit den folgenden Pragmas kann die Variable a nur mit Schreibrecht und Variable b überhaupt nicht in die Symboldatei exportiert werden:

```
VAR
   a : INT {flag noread};
   b : INT {flag noread, nowrite};
END_VAR
VAR
   { flag noread on}
   a : INT;
   { flag noread, nowrite on}
   b : INT;
   {flag off}
END_VAR
```

Beispiel 2

Beide Variablen a und b werden nicht in die Symboldatei exportiert:

```
{ flag noread, nowrite on }
VAR
   a : INT;
   b : INT;
END_VAR
{flag off}

VAR
   { flag noread, nowrite on }
   a : INT;
   b : INT;
   {flag off}
END_VAR
```

Pragma wirkt additiv auf alle untergeordneten Variablendeklarationen

Beispiel

Alle Bausteine werden mit Lese- und Schreibrecht exportiert.

```
a : afb;
...
FUNCTION_BLOCK afb
   VAR
      b : bfb {flag nowrite};
      c : INT;
   END_VAR
...
FUNCTION_BLOCK bfb
   VAR
      d : INT {flag noread};
      e : INT {flag nowrite};
   END_VAR
```

Die Variable `a.b.d` wird nicht exportiert.

Die Variable `a.b.e` wird nur mit Leserecht exportiert.

Die Variable `a.c` wird mit Lese- und Schreibrecht exportiert.

Pragma {bitaccess}

Das Pragma `{bitaccess}` wird für den Bit-Zugriff verwendet.

Mit dem Pragma `{bitaccess}` können Sie gültige symbolische Bit-Zugriffe auf Struk-

turen definieren. Die Bit-Zugriffe erfolgen mit Hilfe einer globalen Konstanten. Die Symbole sind dann in der Eingabehilfe und in der **Intellisense**-Funktion verfügbar und werden für die Darstellung der Bit-Zugriffe beim Überwachen (Monitoring) im Deklarationseditor verwendet. Die verwendeten globalen Konstanten werden ebenfalls im Deklarationseditor angezeigt.

Hinweis:

Damit Sie das Pragma `{bitaccess}` verwenden können, müssen Sie in den Projektoptionen, Kategorie [Übersetzungsoptionen] (Build]) die Option [Konstanten ersetzen] ([Replace constants]) aktivieren.

Das Pragma müssen Sie in der Definition der Struktur in eine separate Zeile einfügen werden. Die Zeile wird nicht durch einen Strichpunkt abgeschlossen.

Syntax

`{bitaccess <Globale Konstante> <Bitnummer> '<Kommentar>'}`

`<Globale Konstante>`: Name der globalen Konstanten, die in einer globalen Variablenliste definiert sein muss.

`<Bitnummer>`: Wert der globalen Konstanten, wie in der globalen Variablenliste definiert.

Ein Beispiel finden Sie auf Seite 384 (→ Kap. 8.2.4, Adressierung von Bits in Variablen).

5.3.3 Pragmas für Anzeige/Nicht-Anzeige von Deklarationsteilen im Bibliotheksverwalter

Mit Hilfe der Pragmas `{library public}` und `{library private}` können Sie in einer Bibliothek definieren, welche Zeilen/Zeilenteile des Deklarationsteils später bei der Verwendung der Bibliothek in einem Projekt im Bibliotheksverwalter angezeigt bzw. nicht angezeigt werden.

Damit können Sie Kommentare oder bestimmte Variablendeklarationen der Bibliothek für den Benutzer unsichtbar machen. Die Pragmas `{library private}` und `{library public}` gelten jeweils für den Rest derselben bzw. die nachfolgenden Zeilen, solange, bis sie durch das jeweils andere Pragma aufgehoben werden.

Syntax

`{library public}`

Der nachfolgende Text wird im Bibliotheksverwalter angezeigt.

`{library private}`

Der nachfolgende Text wird nicht im Bibliotheksverwalter angezeigt.

Beispiel

Deklarationsteil einer Bibliothek, die im PLC-Editor erstellt wird.

Der Kommentar `(* for all *)` soll nach dem Einbinden der Bibliothek im Bibliotheksverwalter angezeigt werden, der Kommentar `(* not for all *)`jedoch nicht.

Die Variablen `local` und `in3` sollen ebenfalls nicht sichtbar sein.

```
{library public}(* for all *) {library private} (* not for all *)
{library public}
FUNCTION afun : BOOL
```

```
VAR_INPUT
   in: BOOL;
END_VAR
{library private}
VAR
   local: BOOL;
END_VAR
{library public}
VAR_INPUT
   in2: BOOL;
   {library private}
   in3: BOOL;
   {library public}
END_VAR
```

5.4 Editoren der textorientierten Programmiersprachen

5.4.1 Allgemeines zu Texteditoren

Die Texteditoren, die für den Implementierungsteil verwendet werden, der Anweisungslisten-Editor und der Editor für Strukurierten Text, verfügen über die üblichen Funktionalitäten der Windows-Texteditoren.

Die Implementierung in den Texteditoren wird durch **Syntaxcoloring** unterstützt.

Wenn Sie im Überschreibmodus arbeiten, wird in der Statusleiste [ÜB] ([OV]) schwarz angezeigt. Mit der <Einfg>-Taste können Sie zwischen dem Überschreib- und dem Einfügemodus wechseln.

Die wichtigsten Befehle finden Sie im Kontextmenü, das Sie durch einen Klick mit der rechten Maustaste öffnen.

Weitere Informationen finden Sie in den folgenden Kapiteln:

- (→ Kap. 5.4.2, [Einfügen] / [Operator] in Texteditoren), Seite 217
- (→ Kap. 5.4.3, [Einfügen] / [Operand] in Texteditoren), Seite 217
- (→ Kap. 5.4.4, [Einfügen] / [Funktion] in Texteditoren), Seite 218
- (→ Kap. 5.4.5, [Einfügen] / [Funktionsblock] in Texteditoren), Seite 218
- (→ Kap. 5.4.6, Bausteinaufruf mit Ausgangsparametern in Texteditoren), Seite 218
- (→ Kap. 5.4.7, Die Texteditoren im Online-Modus), Seite 219
- (→ Kap. 5.4.8, [Extras] / [Monitoring Einstellungen] ([Extras] / [Monitoring Options])), Seite 219
- (→ Kap. 5.4.9, Breakpoint-Positionen im Texteditor), Seite 220
- (→ Kap. 5.4.10, Was passiert an einem Breakpoint), Seite 221
- (→ Kap. 5.4.11, Zeilennummern des Texteditors), Seite 221
- (→ Kap. 5.4.12, Der Anweisungslisten-Editor AWL (IL)), Seite 221
- (→ Kap. 5.4.13, Der Editor für Strukturierten Text ST), Seite 222

5.4.2 [Einfügen] / [Operator] in Texteditoren

Mit dem Menübefehl [Einfügen] / [Operator] ([Insert] / [Operator]) fügen Sie einen Operator ein.

Nach dem Ausführen des Menübefehls wird ein Dialog geöffnet, der alle Operatoren anzeigt, die in der aktuellen Sprache verfügbar sind.

Wählen Sie einen Operator aus und schließen Sie den Dialog mit [OK]. Der ausgewählte Operator wird an der aktuellen Cursorposition eingefügt. Die Handhabung erfolgt ähnlich wie bei der **Eingabehilfe**.

5.4.3 [Einfügen] / [Operand] in Texteditoren

Mit dem Menübefehl [Einfügen] / [Operand] ([Insert] / [Operand]) fügen Sie einen Operanden ein.

Nach dem Ausführen des Menübefehls wird ein Dialog angezeigt, der alle Variablen anzeigt, die als Operanden zur Verfügung stehen.

Sie können unter folgenden Variablenlisten wählen:

- globale Variablen
- lokale Variablen
- Systemvariablen

Wählen Sie einen Operanden aus und schließen Sie den Dialog mit [OK]. Der ausgewählte Operand wird an der aktuellen Cursorposition eingefügt. Die Handhabung erfolgt ähnlich wie bei der **Eingabehilfe**.

5.4.4 [Einfügen] / [Funktion] in Texteditoren

Mit dem Menübefehl [Einfügen] / [Funktion] ([Insert] / [Function]) fügen Sie eine Funktion ein.

Nach dem Ausführen des Menübefehls wird ein Dialog geöffnet, der alle Funktionen anzeigt.

Sie können unter folgenden Funktionslisten wählen:

- benutzerdefinierten Funktionen
- Standardfunktionen

Wählen Sie eine Funktionen aus und schließen Sie den Dialog mit [OK]. Die ausgewählte Funktion wird an der aktuellen Cursorposition eingefügt. Die Handhabung erfolgt ähnlich wie bei der **Eingabehilfe**.

Wenn Sie im Dialog die Option [Mit Argumenten] ([With Arguments]) aktiviert haben, werden die erforderlichen Eingabevariablen der Funktion ebenfalls eingefügt.

5.4.5 [Einfügen] / [Funktionsblock] in Texteditoren

Mit dem Menübefehl [Einfügen] / [Funktionsblock] ([Insert] / [Function Block]) fügen Sie einen Funktionsblock ein.

Sie können unter folgenden Funktionsblocklisten wählen:

- benutzerdefinierten Funktionsblöcke
- Standardfunktionsblöcke

Wählen Sie einen Funktionsblock aus und schließen Sie den Dialog mit [OK]. Der ausgewählte Funktionsblock wird an der aktuellen Cursorposition eingefügt. Die Handhabung erfolgt ähnlich wie bei der **Eingabehilfe**.

Wenn Sie im Dialog die Option [Mit Argumenten] ([With Arguments]) aktiviert haben, werden die erforderlichen Eingabevariablen des Funktionsblocks ebenfalls eingefügt. Die Variablen müssen Sie jedoch nicht zwingend belegen.

5.4.6 Bausteinaufruf mit Ausgangsparametern in Texteditoren

In den textorientierten Programmiersprachen AWL und ST können Sie die Ausgangsparameter eines aufgerufenen Bausteins direkt beim Aufruf zuweisen.

Beispiel

Der Ausgangsparameter `out1` des Bausteins `afbinst` wird der Variablen a zugewiesen.

Beispiel in AWL

```
CAL afbinst(in1:=1, out1=>a)
```

Beispiel in ST

```
afbinst(in1:=1, out1=>a);
```

Wenn Sie den Baustein unter Verwendung der Eingabehilfe (<F2>-Taste) mit der Option [Mit Argumenten] ([With Arguments]) im Implementationsfenster eines ST oder AWL-Bausteins einfügen, wird er automatisch in dieser Syntax mit seinen Parametern dargestellt. Die Parameter müssen Sie jedoch nicht zwingend belegen.

5.4.7 Die Texteditoren im Online-Modus

Die Editoren verfügen über die Online-Funktionen **Breakpoint setzen** und **Einzelschrittabarbeitung (Steppen)**. In Verbindung mit der Überwachungsfunktion (Monitoring) steht Ihnen so die **Debugging**-Funktionalität moderner Windows-Hochsprachen zur Verfügung.

Im Online-Modus ist das Texteditor-Fenster zweigeteilt. Auf der linken Seite des Fensters befindet sich der normale Programmtext. Auf der rechten Seite werden die Variablen dargestellt, deren Werte in der jeweiligen Zeile geändert werden.

Die Darstellung ist dieselbe wie die im Deklarationsteil. Wenn die Steuerung läuft, werden die momentanen Werte der jeweiligen Variablen dargestellt.

Beachten Sie Folgendes beim Überwachen (Monitoring) von Ausdrücken oder bitadressierten Variablen:

- Bei Ausdrücken wird stets der Wert des gesamten Ausdrucks dargestellt.

  Beispiel: a `AND` b wird blau bzw. mit `:= TRUE` dargestellt, wenn a und b TRUE sind.

- Bei bit-adressierten Variablen wird immer der angesprochene Bit-Wert überwacht.

  Beispiel: a`.`3 wird blau bzw. mit `:= TRUE` dargestellt, wenn a den Wert "4" hat.

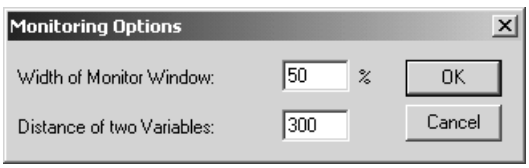
Wenn Sie den Mauszeiger eine kurze Zeit über einer Variablen halten, werden der Typ, die Adresse und der Kommentar der Variablen in einem Tooltip angezeigt.

5.4.8 [Extras] / [Monitoring Einstellungen] ([Extras] / [Monitoring Options])

Mit dem Menübefehl [Extras] / [Monitoring Einstellungen] konfigurieren Sie das Monitoring-Fenster. In den Texteditoren ist das Fenster beim Überwachen (Monitoring) zweigeteilt. Auf der linke Hälfte steht das Programm. Auf der rechte Hälfte finden Sie alle Variablen, die in der entsprechenden Programmzeile stehen.

Mit der Option [Breite des Monitoringfenster] ([Width of Monitor Window]) stellen Sie die Breite des Monitoring-Bereichs im Textfenster ein.

Mit der Option [Abstand zweier Variablen] ([Distance of two Variables]) stellen Sie den Abstand zwischen zwei Monitoring-Variablen in einer Zeile ein. Die Abstandsangabe "1" entspricht dabei der Zeilenhöhe der gewählten Schriftart.



20403AEN

5.4.9 Breakpoint-Positionen im Texteditor

Breakpoint-Positionen sind alle Stellen im Programm, an denen sich Variablenwerte ändern können oder an denen der Programmfluss verzweigt ist. Eine Ausnahme sind Funktionsaufrufe, bei denen gegebenenfalls ein Breakpoint in der Funktion gesetzt werden muss. An den dazwischen liegenden Positionen ist ein Breakpoint nicht sinnvoll, weil sich an den Daten seit der vorhergehenden Breakpoint-Position nichts geändert haben kann.

Da im PLC-Editor intern mehrere AWL-Zeilen zu einer C-Code-Zeile zusammengefasst werden, können nicht in jeder Zeile Breakpoints gesetzt werden.

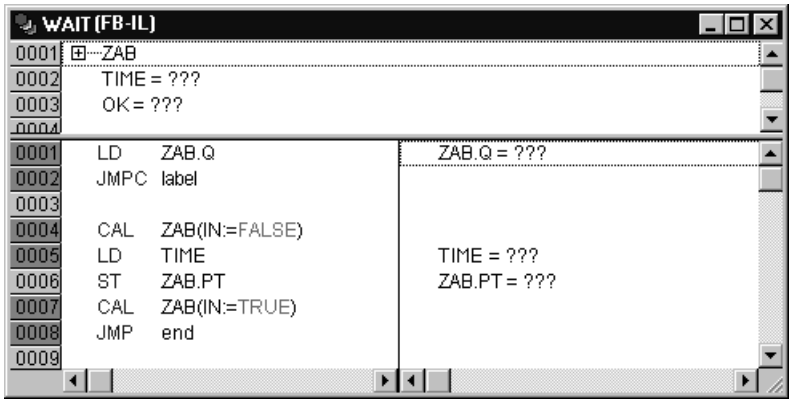
Damit ergeben sich in AWL folgende Breakpoint-Positionen:

- am Anfang des Bausteins
- bei jedem `LD-`, `LDN`-Operator (oder, falls ein `LD`-Operator direkt auf eine Marke folgt, auf dieser Marke)
- bei jedem `JMP-` , `JMPC-` , `JMPCN`-Operator
- bei jeder Marke
- bei jedem `CAL-` , `CALC-` , `CALCN`-Operator
- bei jedem `RET-` , `RETC-` , `RETCN`-Operator
- am Ende des Bausteins

Für Strukturierten Text ergeben sich folgende Breakpoint-Positionen:

- bei jeder Zuweisung
- bei jeder `RETURN`- und `EXIT`-Anweisung
- in Zeilen, in denen Bedingungen ausgewertet werden (`WHILE`, `IF`, `REPEAT`)
- am Ende des Bausteins

Breakpoint-Positionen erkennen Sie daran, dass der Hintergrund des Zeilennummernfeldes dunkelgrau ist.



20425AEN

Breakpoint aktivieren

Um einen Breakpoint zu setzen, klicken Sie mit der Maus auf das Zeilennummernfeld der Zeile, in der Sie den Breakpoint setzen möchten. Wenn das ausgewählte Feld eine Breakpoint-Position ist, so wechselt die Farbe des Zeilennummernfeldes von dunkelgrau nach hellblau und der Breakpoint wird in der Steuerung aktiviert.

Wenn Sie Breakpoints aktivieren bzw. deaktivieren, aber nicht löschen möchten, führen Sie einen der folgenden Schritte aus

- Klicken Sie in die Zeile mit dem Breakpoint und führen Sie den Menübefehl [Online] / [Breakpoint an /aus] ([Online] / [Toggle Breakpoint]) aus. Alternativ können Sie die <F9>-Taste oder die Schaltfläche [ ] in der Funktionsleiste benutzen.
- Klicken Sie auf das Zeilennummernfeld der Zeile mit dem Breakpoint.

Breakpoint löschen

Um einen Breakpoint zu löschen, aktivieren Sie zuerst die zu löschenden Breakpoints.

Führen Sie anschließend den Menübefehl [Online] / [Breakpoint-Dialog] ([Online] / [Breakpoint Dialog]).

Der Dialog [Breakpoints] ([Breakpoints]) wird geöffnet. In dem Dialog sind die aktivierten Breakpoints aufgelistet. Klicken Sie auf einen Breakpoint und anschließend auf die Schaltfläche [Löschen] ([Delete]). Um alle Breakpoints in der Liste zu löschen, klicken Sie auf die Schaltfläche [Alle löschen] ([Delete all]).

5.4.10 Was passiert an einem Breakpoint

Wenn der Programmablauf der Steuerung einen Breakpoint erreicht, wird am Bildschirm der Ausschnitt mit der entsprechenden Zeile dargestellt. Das Zeilennummernfeld der Zeile, in der die Steuerung steht, wird rot markiert. Die Steuerung stoppt die Bearbeitung des Anwenderprogramms.

Wenn das Programm auf einem Breakpoint steht, können Sie die Bearbeitung mit dem Befehl [Online] / [Start] ([Online] / [Run]) fortsetzen. Außerdem können Sie mit dem Befehl [Online] / [Einzelschritt über] ([Online] / [Step over]) oder [Einzelschritt in] ([Step in]) nur bis zur nächsten Breakpoint-Position springen.

Wenn die Anweisung, auf der das Programm steht, ein CAL-Befehl ist oder wenn in den Zeilen bis zur nächsten Breakpoint-Position ein Funktionsaufruf steht, wird dieser CAL-Befehl bzw. Funktionsaufruf mit dem Befehl [Einzelschritt über] ([Step over]) übersprungen. Mit dem Menübefehl [Einzelschritt in] ([Step in]) wird in den aufgerufenen Baustein verzweigt.

5.4.11 Zeilennummern des Texteditors

Die Zeilennummern des Texteditors geben die Textzeilennummer einer Baustein-Implementierung an.

Im Offline-Modus markieren Sie durch einen einfachen Klick auf eine spezielle Zeilennummer die gesamte Textzeile.

Im Online-Modus zeigt die Hintergrundfarbe der Zeilennummer den Breakpoint-Zustand der Zeile an:

- dunkelgrau: Diese Zeile ist eine mögliche Position für einen Breakpoint.
- hellblau: In dieser Zeile wurde ein Breakpoint gesetzt.
- rot: Die Programmabarbeitung befindet sich an diesem Punkt.

Im Online-Modus wechseln Sie durch einen einfachen Mausklick auf die Zeilennummer den Breakpoint-Zustand der Zeile.

5.4.12 Der Anweisungslisten-Editor AWL (IL)

Der Anweisungslisten-Editor

Die Editoren für die Bausteine bestehen aus einem Deklarationsteil und einem Rumpf. Die Bildschirmdarstellung der Editoren ist durch einen Bildschirmteiler geteilt.

Der Anweisungslisten-Editor AWL ist ein Texteditor mit den üblichen Funktionalitäten der Windows-Texteditoren.

Die wichtigsten Befehle finden Sie im Kontextmenü, das Sie durch einen Klick mit der rechten Maustaste öffnen.

Im Anweisungslisten-Editor AWL ist ein mehrzeiliger Baustein-Aufruf zulässig.

Beispiel

```
CAL CTU_inst(
   CU:=%IX10,
   PV:=(
      LD A
      ADD 5
   )
)
```

Nähere Informationen zur Programmiersprache AWL finden Sie auf Seite 24 (→ Kap. 2.4.2, Anweisungsliste AWL (IL)).

AWL im Online-Modus

Mit dem Menübefehl [Online] / [Ablaufkontrolle] ([Online] / [Display Flow Control]) fügen Sie im AWL-Editor auf der linken Seite jeder Zeile ein Feld ein, das den Akkumulatorinhalt darstellt.

Nähere Informationen zum AWL-Editor im Online-Modus finden Sie auf Seite 219 (→ Kap. 5.4.7, Die Texteditoren im Online-Modus).

5.4.13 Der Editor für Strukturierten Text ST

Die Editoren für die Bausteine bestehen aus einem Deklarationsteil und einem Rumpf. Die Bildschirmdarstellung der Editoren ist durch einen Bildschirmteiler geteilt.

Der Editor für Strukturierten Text ist ein Texteditor mit den üblichen Funktionalitäten der Windows-Texteditoren.

Die wichtigsten Befehle finden Sie im Kontextmenü, das Sie durch einen Klick mit der rechten Maustaste öffnen.

Nähere Informationen zum Editor für Strukturierten Text im Online-Modus finden Sie auf Seite 219 (→ Kap. 5.4.7, Die Texteditoren im Online-Modus).

Nähere Informationen zur Programmiersprache Strukturierter Text finden Sie auf Seite 26 (→ Kap. 2.4.3, Strukturierter Text ST (ST)).

5.5 Die Editoren der grafisch orientierten Programmiersprachen

5.5.1 Die grafischen Editoren

Editoren für folgende grafisch orientierten Programmiersprachen stehen zur Verfügung:

- Ablaufsprache AS (SFC)
- Kontaktplan KOP (LD)
- Funktionsplan FUP (FBD)
- Freigraphischer Funktionsplan CFC

Die Editoren haben viele Gemeinsamkeiten, die in den folgenden Kapiteln zusammengefasst werden:

- (→ Kap. 5.5.2, Zoom), Seite 223
- (→ Kap. 5.5.3, Netzwerk), Seite 224
- (→ Kap. 5.5.4, Sprungmarken), Seite 224
- (→ Kap. 5.5.5, Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]), Seite 224
- (→ Kap. 5.5.6, [Einfügen] / [Netzwerk (danach)] oder [Einfügen] / [Netzwerk (davor)]), Seite 227
- (→ Kap. 5.5.7, Die Netzwerk-Editoren im Online-Modus), Seite 227

Zusätzlich gibt es die spezifischen Beschreibungen der grafischen Editoren.

Die Implementierung in den grafischen Editoren wird durch die Funktion **Syntaxcoloring** unterstützt.

Weitere Informationen finden Sie in den folgenden Kapiteln:

- (→ Kap. 5.5.2, Zoom), Seite 223
- (→ Kap. 5.5.3, Netzwerk), Seite 224
- (→ Kap. 5.5.4, Sprungmarken), Seite 224
- (→ Kap. 5.5.5, Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]), Seite 224
- (→ Kap. 5.5.6, [Einfügen] / [Netzwerk (danach)] oder [Einfügen] / [Netzwerk (davor)]), Seite 227
- (→ Kap. 5.5.7, Die Netzwerk-Editoren im Online-Modus), Seite 227
- (→ Kap. 5.5.8, Der Funktionsplan-Editor FUP (FBD)), Seite 229
- (→ Kap. 5.5.9, Der Kontaktplan-Editor KOP (LD)), Seite 237
- (→ Kap. 5.5.10, Der Ablaufsprachen-Editor AS (SFC)), Seite 244
- (→ Kap. 5.5.11, Der Freigrafische Funktionsplan-Editor CFC), Seite 256

5.5.2 Zoom

Objekte wie Bausteine, Aktionen, Transitionen etc. können Sie in den Sprachen AS, KOP, FUP, CFC und in Visualisierungen mit einer Zoom-Funktion vergrößern oder verkleinern. Dabei werden alle Elemente des Fensterinhalts des Implementationsteils erfasst, der Deklarationsteil bleibt unverändert.

Standardmäßig wird jedes Objekt mit der Zoomstufe 100% angezeigt. Die eingestellte Zoomstufe wird als Objekteigenschaft im Projekt abgespeichert.

Das Ausdrucken der Projektdokumentation erfolgt immer in der Darstellung 100%!

Die Zoomstufe können Sie mit Hilfe der Auswahlliste in der Symbolleiste einstellen. Sie können Werte zwischen 25% und 400% auswählen. Individuelle Werte zwischen 10%

und 500% können Sie manuell eingeben.

Die Auswahl der Zoomstufe steht Ihnen nur zur Verfügung, wenn der Cursor in einem in einer grafisch orientierten Programmiersprache erstellten Objekt oder in einem Visualisierungsobjekt steht.

Die Cursorposition in den Editoren können Sie auch im gezoomten Zustand des Objekts auswählen. Die Cursorposition können Sie mit den Pfeiltasten ändern. Die dargestellte Textgröße hängt vom Zoomfaktor und der eingestellten Schriftgröße ab.

Die Ausführung aller Menüpunkte zur Bedienung des Editors entsprechend der Cursorposition ist bei jeder Zoomstufe und unter Beibehaltung der Zoomstufe möglich.

Im Online-Modus wird jedes Objekt entsprechend der eingestellten Zoomstufe dargestellt. Die Online-Funktionalitäten sind uneingeschränkt verfügbar.

Bei Verwendung der IntelliMouse können Sie ein Objekt vergrößern bzw. verkleinern, indem Sie die <Strg>-Taste drücken und gleichzeitig das Rad vorwärts bzw. rückwärts drehen.

5.5.3 Netzwerk

In den Programmiersprachen KOP und FUP wird das Programm in einer Liste von Netzwerken angeordnet.

Jedes Netzwerk ist auf der linken Seite mit einer fortlaufenden Netzwerknummer gekennzeichnet und enthält eine Struktur, die Folgendes darstellt kann:

- einen logischen bzw. arithmetischen Ausdruck
- einen Programmaufruf
- einen Funktionsaufruf
- einen Funktionsblockaufruf
- einen Sprung
- eine `Return`-Anweisung

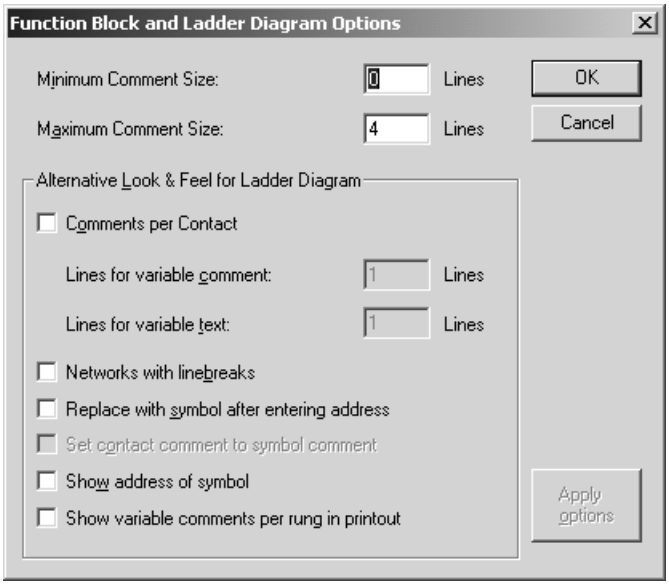
5.5.4 Sprungmarken

Jedes Netzwerk verfügt über eine Sprungmarke, die wahlweise auch leer sein kann. Diese Sprungmarke bearbeiten Sie, indem Sie in die erste Zeile des Netzwerks, unmittelbar neben die Netzwerknummer klicken. Anschließend geben Sie die Sprungmarke und einen Doppelpunkt ein.

5.5.5 Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]

Zu jedem Netzwerk können Sie einen mehrzeiligen Kommentar eingeben.

Im Dialog [Funktions- und Kontaktplan Optionen] ([Funktion Block and Ladder Diagram Options]), den Sie mit dem Befehl [Extras] / [Optionen] ([Extras] / [Options]) öffnen, können Sie Optionen der Kommentare einstellen:

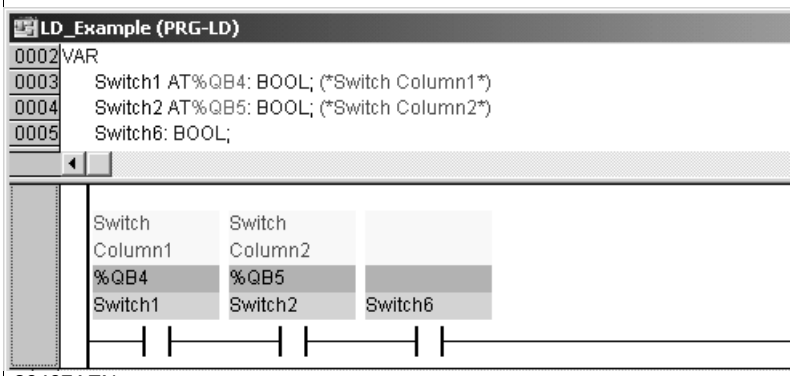
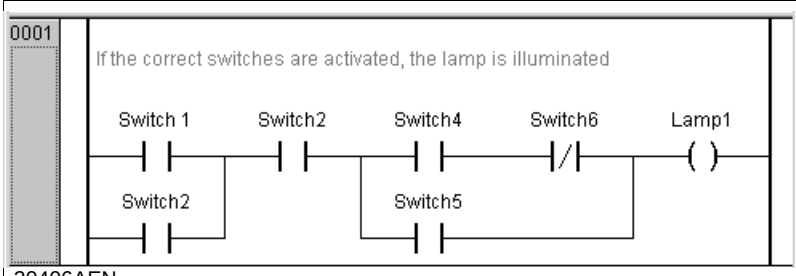


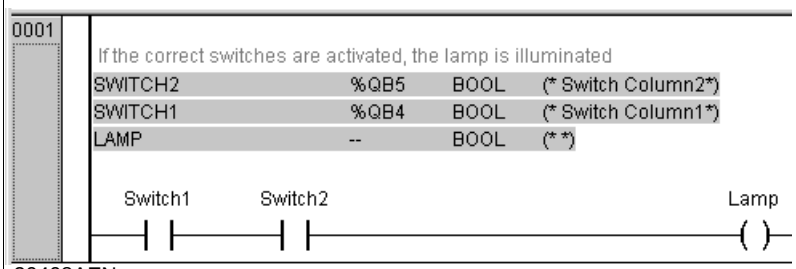
20405AEN

[Minimale Kommentargröße]	[Minimum Comment Size]	Anzahl der Zeilen, die generell für Kommentare freigelassen bzw. angezeigt werden. **Beispiel:** Ist diese Option auf 2 eingestellt, stehen an jedem Netzwerkanfang nach der Labelzeile zwei leere Kommentarzeilen. (Der Vorgabewert "0" hat den Vorteil hat, dass mehrere Netzwerke in den Bildschirmbereich passen.) Wenn diese Option größer als 0 ist, können Sie einen Kommentar eingeben indem Sie einfach in die angezeigte Kommentarzeile klicken und den Kommentartext eingeben. Andernfalls müssen Sie zunächst das Netzwerk, zu dem Sie einen Kommentar eingeben möchten, auswählen und mit dem Befehl [Einfügen] / [Kommentar] ([Insert] / [Comment]) eine Kommentarzeile einfügen. Kommentare werden im Unterschied zum Programmtext grau dargestellt.
[Maximale Kommentargröße]	[Maximum Comment Size]	Maximale Anzahl der Zeilen, die für einen Netzwerkkommentar zur Verfügung stehen. (Vorgabewert: 4).

[Alternatives Look & Feel] ([Alternative Lock & Feel for Ladder Diagram]):

Die folgenden Optionen ermöglichen eine alternative Darstellung der Netzwerke.

[Kommentare pro Kontakt] (nur für Kontaktplan)	[Comments per Contact]	Wenn diese Option aktiviert ist, können Sie Kommentare für einzelne Kontakte und Spulen eingeben. Geben Sie die gewünschte Anzahl von Zeilen, die dafür vorgesehen und angezeigt werden soll, im Feld [Zeilen für Variablenkommentar] ([Lines for variable comment]) ein. Daraufhin erscheint über dem Kontakt bzw. der Spule ein Kommentarfeld, in dem Sie den Kommentartext eingeben. Wenn die Option [Kommentare pro Kontakt] ([Lines per contact]) aktiviert ist, können Sie im Feld [Zeilen für Variablentext] ([Lines for variable text]) die Anzahl der Zeilen definieren, die für den Variablennamen des Kontakts bzw. der Spule verwendet wird. Somit können auch lange Namen durch die Verwendung von mehreren Zeilen komplett dargestellt werden. Im folgenden Beispiel sind zwei Zeilen für den Kontaktkommentar und eine Zeile für den Variablentext vorgesehen:
	 20407AEN	
[Netzwerke mit Umbrüchen] (nur für Kontaktplan)	[Networks with linebreaks]	Wenn diese Option aktiviert ist, werden in die Netzwerke Umbrüche eingefügt, sobald die eingestellte Fensterbreite nicht mehr erlaubt, dass alle Elemente des Netzwerks sichtbar sind.
	 20406AEN	
[Mit Symbol ersetzen, nach Eingabe der Adresse]	[Replace with symbol after entering address]	Wenn diese Option aktiviert ist, können Sie am Baustein bzw. am Kontakt oder Spule eine Adresse (z. B. %QB4) eingeben. Diese Adresse wird unmittelbar nach der Eingabe durch den Namen der Variablen ersetzt, die dieser Adresse zugewiesen ist. Wenn einer Adresse keine Variable zugewiesen ist, wird die Anzeige der Adresse nicht verändert.
[Kontaktkommentar mit Symbolkommentar vorbesetzen] (nur für Kontaktplan)	[Set contact comment to symbol comment]	Wenn diese Option aktiviert ist, wird im Kommentarfeld des Kontakts oder der Spule der Kommentar angezeigt, der für die verwendete Variable definiert wurde. Der Kommentar kann dort weiter bearbeitet werden (siehe Beispiel in Abbildung oben bei [Kommentare pro Kontakt]). Dazu muss jedoch die Option [Kommentare pro Kontakt] ([Comments per contact]) aktiviert sein. **Beachten Sie:** Wenn in der Variablendeklaration kein Kommentar vorliegt, wird in diesem Fall ein im Kommentarfeld bereits lokal eingetragener Kommentar automatisch durch den Variablenkommentar ersetzt, ggf. durch Leerzeichen!

[Adresse des Symbols anzeigen] (nur für Kontaktplan)	[Show address of symbol]	Wenn die am Kontakt bzw. an der Spule eingetragene Variable einer Adresse zugewiesen ist, wird diese zusätzlich oberhalb des Variablennamens angezeigt (siehe folgende Abbildung).
[Variablenkommentare pro Netzwerk im Ausdruck anzeigen]	[Show variable comments per rung in printout]	Wenn diese Option aktiviert ist, wird pro Netzwerk für jede im Netzwerk verwendete Variable eine Zeile angezeigt. Diese Anzeige der Variable besteht aus dem Variablennamen, der Adresse, dem Datentyp und dem Variablenkommentar, der in der Variablendeklaration definiert wurde. Dies kann für die Dokumentation des Projekts (z. B. beim Ausdrucken) von Nutzen sein.
		

Anwenden der Optionen

[OK]

Wenn Sie auf die Schaltfläche [OK] klicken, werden die eingestellten Optionen im vorliegenden Baustein angewendet und der Dialog geschlossen.

[Optionen anwenden]

Wenn Sie auf die Schaltfläche [Optionen anwenden] ([Apply options]) klicken, werden die eingestellten Optionen im gesamten Projekt angewendet. Es erscheint ein Dialog, in dem Sie dies nochmals explizit bestätigen müssen.

5.5.6 [Einfügen] / [Netzwerk (danach)] oder [Einfügen] / [Netzwerk (davor)]

Tastaturbefehl: <Umschalt>+<T>

Um ein neues Netzwerk im FUP- oder KOP-Editor vor dem aktuellen Netzwerk einzufügen, wählen Sie den Befehl [Einfügen] / [Netzwerk (davor)] ([Insert] / [Network before]).

Um ein neues Netzwerk im FUP- oder KOP-Editor nach dem aktuellen Netzwerk einzufügen, wählen Sie den Befehl [Einfügen] / [Netzwerk (danach)] ([Insert] / [Network after]).

Das aktuelle Netzwerk, gekennzeichnet durch ein gepunktetes Rechteck unter der Netzwerknummer, ändern Sie durch einen Mausklick auf die Netzwerknummer.

Sie können auch einen Bereich von mehreren Netzwerken auswählen, indem Sie die <Umschalt>-Taste drücken und auf ein beliebiges Netzwerk klicken. Dabei wählen Sie den Bereich von Netzwerken zwischen dem aktuellen und dem angeklickten Netzwerk aus.

5.5.7 Die Netzwerk-Editoren im Online-Modus

In den Editoren FUP und KOP können Sie **Breakpoints** nur auf die Netzwerke setzen. Das Netzwerknummernfeld eines Netzwerks, auf das ein Breakpoint gesetzt wurde, wird blau dargestellt. Die Bearbeitung stoppt dann vor dem Netzwerk, auf dem der Breakpoint steht und das Netzwerknummernfeld wird rot dargestellt. Bei der Einzelschrittabarbeitung (Steppen) wird von Netzwerk zu Netzwerk gesprungen.

Alle Werte werden an den Ein- und Ausgängen der Netzwerkbausteine überwacht (Monitoring).

Hinweis:

Beim Überwachen (Monitoring) von Ausdrücken oder bit-adressierten Variablen müssen Sie Folgendes beachten:

Bei Ausdrücken, z. B. a `AND` b als Transitionsbedingung oder Funktionsblockseingang, wird stets der Wert des gesamten Ausdrucks dargestellt (a `AND` b wird blau bzw. mit `:=` `TRUE` angezeigt, wenn a und b gleich TRUE sind).

Bei bit-adressierten Variablen wird immer der angesprochene Bit-Wert überwacht (monitored) (z. B. wird a`.`3 blau bzw. mit `:=` `TRUE` dargestellt, wenn a gleich 4 ist).

Die Ablaufkontrolle starten Sie mit dem Menübefehl [Online] / [Ablaufkontrolle] ([Online] / [Display Flow Control]). Mit Hilfe der Ablaufkontrolle können Sie die aktuellen Werte, die in den Netzwerken über die Verbindungslinien transportiert werden, ansehen.

- Wenn die Verbindungslinien keine booleschen Werte transportieren, wird der Wert in einem separat eingefügten Feld angezeigt. Die Monitorfelder für Variablen, die nicht verwendet werden (z. B. bei der Funktion `SEL`) werden grau schattiert dargestellt.
- Wenn die Verbindungslinien boolesche Werte übertragen, werden die Verbindungslinien blau dargestellt, wenn deren Wert auf TRUE gesetzt ist.

So können Sie den Informationsfluss während des Steuerungsablaufs mit verfolgen.

Wenn Sie den Mauszeiger eine kurze Zeit über einer Variablen halten, wird der Typ, die Adresse und der Kommentar der Variablen in einem Tooltip angezeigt.

5.5.8 Der Funktionsplan-Editor FUP (FBD)

Überblick

Der Funktionsplan-Editor ist ein grafischer Editor, der mit einer Liste von Netzwerken arbeitet. Jedes Netzwerk enthält eine Struktur, die Folgendes darstellen kann:

- einen logischen bzw. arithmetischen Ausdruck
- einen Programmaufruf
- einen Funktionsaufruf
- einen Funktionsblockaufruf
- einen Sprung
- eine `Return`-Anweisung

Nähere Informationen über die Sprache finden Sie auf Seite 40 (→ Kap. 2.4.5, Funktionsplan FUP (FBD)).

Weitere Informationen finden Sie in den folgenden Abschnitten:

- (→ Abschnitt. Cursorpositionen im FUP), Seite 229
- (→ Abschnitt. Wie Sie im FUP den Cursor setzen), Seite 230
- (→ Abschnitt. [Einfügen] / [Zuweisung] im FUP), Seite 231
- (→ Abschnitt. [Einfügen] / [Sprung] im FUP), Seite 231
- (→ Abschnitt. [Einfügen] / [Return] im FUP), Seite 232
- (→ Abschnitt. [Einfügen] / [Baustein] im FUP), Seite 232
- (→ Abschnitt. [Einfügen] / [Eingang] im FUP), Seite 233
- (→ Abschnitt. [Einfügen] / [Ausgang] im FUP), Seite 233
- (→ Abschnitt. [Extras] / [Negation] im FUP), Seite 234
- (→ Abschnitt. [Extras] / [Set/Reset] im FUP), Seite 234
- (→ Abschnitt. [Extras] / [Ansicht] im FUP), Seite 234
- (→ Abschnitt. Zoom zum aufgerufenen Baustein), Seite 235
- (→ Abschnitt. [Extras] / [Instanz öffnen]), Seite 197
- (→ Abschnitt. Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]), Seite 224
- (→ Abschnitt. Ausschneiden, Kopieren, Einfügen und Löschen in FUP), Seite 235
- (→ Abschnitt. Der Funktionsplan im Online-Modus), Seite 236

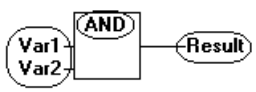
Cursorpositionen im FUP

Jeder Text ist eine mögliche Cursorposition. Der markierte Text ist blau hinterlegt und kann nun geändert werden.

Ansonsten ist die aktuelle Cursorposition durch ein gepunktetes Rechteck gekennzeichnet.

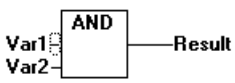
Mögliche Cursorpositionen (schwarz umrahmt bzw. gestrichelt) mit einem Beispiel:

1. Textfelder:



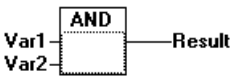
20426AXX

2. Eingänge:



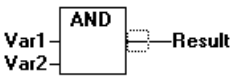
20427AXX

3. Operatoren, Funktionen oder Funktionsbausteine:



20428AXX

4. Ausgänge, wenn danach eine Zuweisung oder ein Sprung folgt:

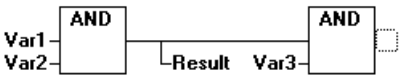


20429AXX

5. Linienkreuz über einer Zuweisung, einem Sprung oder einer `Return`-Anweisung:

20430AXX

6. Hinter dem äußersten rechten Objekt eines jeden Netzwerks ("letzte Cursorposition", dies ist auch die Cursorposition, wenn ein Netzwerk ausgewählt wurde):



20431AXX

7. Linienkreuz unmittelbar vor einer Zuweisung:

20432AXX

Wie Sie im FUP den Cursor setzen

Den Cursor können Sie durch einen einfachen Mausklick oder mit Hilfe der Tastatur auf eine bestimmte Position setzen.

Wenn Sie die Pfeiltasten drücken, springt der Cursor jeweils zur nächstliegenden Cursorposition in der ausgewählten Richtung. Der Cursor kann mit den Pfeiltasten alle Cursorpositionen, einschließlich der Textfelder, erreichen. Wenn die letzte Cursorposition ausgewählt ist, können Sie mit den Pfeiltasten <nach oben> bzw. <nach unten> zur letzten Cursorposition des vorhergehenden bzw. nachfolgenden Netzwerks springen.

Ein leeres Netzwerk enthält drei Fragezeichen "???". Wenn Sie hinter diese Fragezeichen klicken, springt der Cursor auf die letzte Position.

[Einfügen] / [Zuweisung] im FUP

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<A>

Mit dem Menübefehl [Einfügen] / [Zuweisung] ([Insert] / [Assign]) fügen Sie eine Zuweisung ein.

Die Zuweisung- wird abhängig von der aktuellen Cursorposition eingefügt. Weitere Informationen dazu finden Sie auf Seite 229 (→ Abschnitt. Cursorpositionen im FUP).

- unmittelbar vor dem gewählen Eingang
- unmittelbar nach dem gewählten Ausgang
- am Ende des Netzwerks

Nachdem Sie die Zuweisung eingefügt haben, klicken Sie auf die Fragezeichen "???" und ersetzen diese durch die Variable, an die Sie zuweisen wollen. Dazu können Sie auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen.

Weitere Informationen zur möglichen Eingabe einer Adresse anstelle des Variablennamens finden Sie auf Seite 224 (→ Kap. 5.5.5, Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]).

Um zu einer existierenden Zuweisung eine weitere Zuweisung hinzuzufügen, klicken Sie auf den Befehl [Einfügen] / [Ausgang] ([Insert] / [Output]).

[Einfügen] / [Sprung] im FUP

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<L>

Mit dem Menübefehl [Einfügen] / [Sprung] ([Insert] / [Jump]) fügen Sie eine Sprung-Anweisung ein.

Die Sprung-Anweisung wird abhängig von der aktuellen Cursorposition eingefügt. Weitere Informationen dazu finden Sie auf Seite 229 (→ Abschnitt. Cursorpositionen im FUP).

- unmittelbar vor dem gewählten Eingang
- unmittelbar nach dem gewählten Ausgang
- am Ende des Netzwerks

Wenn Sie die Sprung-Anweisung eingefügt haben, können Sie anschließend die Fragezeichen "???" anklicken und durch die Sprungmarke, an die gesprungen werden soll, ersetzen.

[Einfügen] / [Return] im FUP

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<R>

Mit dem Menübefehl [Einfügen] / [Return] ([Insert] / [Return]) fügen Sie eine `Return`-Anweisung ein.

Die `Return`-Anweisung wird abhängig von der aktuellen Cursorposition eingefügt. Weitere Informationen dazu finden Sie auf Seite 229 (→ Abschnitt. Cursorpositionen im FUP).

- unmittelbar vor dem gewählten Eingang
- unmittelbar nach dem gewählten Ausgang
- am Ende des Netzwerks

[Einfügen] / [Baustein] im FUP

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<B>

Mit dem Menübefehl [Einfügen] / [Baustein] ([Insert] / [Box]) fügen Sie Operatoren, Funktionen, Funktionsblöcke und Programme ein. Nach dem Ausführen des Menübefehls wird zunächst ein Operator `AND` eingefügt.

Diesen Operator `AND` können Sie in folgende Konstrukte umwandeln, indem Sie den Text "AND" markieren und überschreiben:

- in einen anderen Operator
- in eine Funktion
- in einen Funktionsblock
- in ein Programm

In der Eingabehilfe, die Sie mit der <F2>-Taste öffnen, können Sie den gewünschten Baustein auswählen.

- Wenn der neue Baustein eine andere Mindestanzahl von Eingängen hat, werden diese angehängt.
- Wenn der neue Baustein eine kleinere Höchstzahl von Eingängen hat, werden die letzten Eingänge gelöscht.

Bei Funktionen und Funktionsblöcken werden die formalen Namen der Ein- und Ausgänge angezeigt.

Bei Funktionsblöcken befindet sich über der Box ein Instanz-Feld, dessen Text Sie ändern können. Wenn Sie einen anderen unbekannten Funktionsblock aufrufen, indem Sie den Text des Types ändern, wird eine Operator-Box mit zwei Eingängen und dem angegeben Typ angezeigt. Wenn das Instanz-Feld angewählt ist, können Sie mit der <F2>-Taste die Eingabehilfe öffnen und die Variablen auswählen.

Der neue Baustein wird abhängig von der aktuellen Cursorposition eingefügt. Weitere Informationen dazu finden Sie auf Seite 229 (→ Abschnitt. Cursorpositionen im FUP).

- Wenn ein Eingang ausgewählt ist, wird der Baustein vor diesem Eingang eingefügt. Der erste Eingang dieses Bausteins wird mit dem Zweig links vom ausgewählten Eingang verbunden. Der Ausgang des neuen Bausteins wird mit dem ausgewählten Eingang verbunden.
- Wenn ein Ausgang ausgewählt ist, wird der Baustein nach diesem Ausgang eingefügt. Der erste Eingang des Bausteins wird mit dem ausgewählten Ausgang verbunden. Der Ausgang des neuen Bausteins wird mit dem Zweig, mit dem der ausge-

wählte Ausgang verbunden war, verbunden.

- Wenn ein Baustein, eine Funktion oder ein Funktionsblock ausgewählt ist, wird das bisherige Element durch den neuen Baustein ersetzt. Die Zweige werden, falls möglich, wie vor der Ersetzung verbunden. Wenn das alte Element mehr Eingänge hatte als das neue Element, dann werden die unverknüpfbaren Zweige gelöscht. Das gleiche gilt für die Ausgänge.
- Wenn ein Sprung oder eine `Return`-Anweisung ausgewählt ist, wird der Baustein vor diesem Sprung, bzw. dieser `Return`-Anweisung eingefügt. Der erste Eingang des Bausteins wird mit dem Zweig links vom ausgewählten Element verbunden. Der Ausgang des Bausteins wird mit dem Zweig rechts vom ausgewählten Element verbunden.
- Wenn die letzte Cursorposition eines Netzwerks ausgewählt ist, wird der Baustein nach dem letzen Element eingefügt. Der erste Eingang des Bausteins wird mit dem Zweig links von der ausgewählten Position verbunden.

Alle Eingänge des Bausteins, die nicht verbunden werden konnten, erhalten einen Standardtext, den durch die gewünschte Konstante bzw. Variable ersetzen müssen.

Wenn rechts von einem eingefügten Baustein ein Ast steht, so wird dieser Ast dem ersten Bausteinausgang zugeordnet. Ansonsten bleiben die Ausgänge unbelegt.

[Einfügen] / [Eingang] im FUP

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<U>

Mit dem Menübefehl [Einfügen] / [Eingang] ([Insert] / [Input]) fügen Sie einen Operatoreingang ein. Die Zahl der Eingänge ist bei vielen Operatoren variabel (z. B. Der Operator ADD kann zwei oder mehrere Eingänge haben).

In Abhängigkeit von der Cursorposition wird der neue Eingang an einer unterschiedlichen Position eingefügt Wenn Sie den neuen Eingang über einem bestehende Eingang einfügen möchten, klicken Sie auf den bestehenden Eingang. Wenn Sie den neuen Eingang ganz unten einfügen möchten, klicken Sie auf den Operator selbst. Weitere Informationen zur Cursorposition finden Sie auf Seite 229 (→ Abschnitt. Cursorpositionen im FUP).

Der eingefügte Eingang ist mit dem Text "???" belegt. Diesen Text müssen Sie anklicken und durch die gewünschte Konstante bzw. Variable ersetzen. Dazu können Sie auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen.

Weitere Informationen zur möglichen Eingabe einer Adresse anstelle des Variablennamens finden Sie auf Seite 224 (→ Kap. 5.5.5, Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]).

[Einfügen] / [Ausgang] im FUP

Schaltfläche: [ ]

Mit dem Menübefehl [Einfügen] / [Ausgang] ([Insert] / [Output]) fügen Sie zu einer Zuweisung eine weitere Zuweisung hinzu. Diese Funktionalität verwenden Sie um sogenannte Zuweisungskämme zu erstellen. Sie weisen mehreren Variablen den aktuellen Wert einer Leitung zu.

- Wenn das Linienkreuz über einer Zuweisung bzw. der unmittelbar davor liegende Ausgang ausgewählt ist, wird nach den bereits vorhandenen Zuweisungen eine weitere Zuweisung eingefügt.
- Wenn das Linienkreuz direkt vor einer Zuweisung ausgewählt ist, wird vor dieser Zu-

weisung eine weitere Zuweisung eingefügt.

Der eingefügte Ausgang ist mit dem Text "???" belegt. Diesen Text müssen Sie anklicken und durch die gewünschte Variable ersetzen. Dazu können Sie auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen.

Weitere Informationen zur möglichen Eingabe einer Adresse anstelle des Variablennamens finden Sie auf Seite 224 (→ Kap. 5.5.5, Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]).

[Extras] / [Negation] im FUP

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<N>

Mit dem Menübefehl [Extras] / [Negation] ([Extras] / [Negate]) negieren Sie das ausgewählte Element. Das Symbol für die Negation ist ein kleiner Kreis auf einer Verbindung.

- Wenn ein Eingang ausgewählt ist, dann wird der Eingang mit diesem Befehl negiert.
- Wenn ein Ausgang ausgewählt ist, dann wird der Ausgang mit diesem Befehl negiert.
- Wenn ein Sprung oder ein Return ausgewählt ist, dann wird der Eingang dieses Sprungs bzw. Returns mit diesem Befehl negiert.

Eine Negation können Sie durch erneutes Negieren löschen.

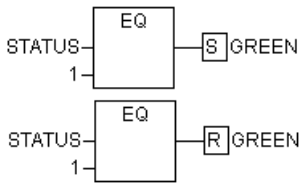
[Extras] / [Set/Reset] im FUP

Schaltfläche: [ ]

Mit dem Menübefehl [Extras] / [Set/Reset] ([Extras] / [Set/Reset]) definieren Sie Ausgänge als Set- bzw. Reset-Ausgänge.

Vor einem Set-Ausgang wird ein [S] dargestellt.

Vor einem Reset-Ausgang wird ein [R] dargestellt.



20433AXX

Ein Set-Ausgang wird auf TRUE gesetzt, wenn der Ausgang des Elements auf TRUE gesetzt wird. Der Set-Ausgang behält diesen Wert, selbst wenn der Ausgang des Elements wieder auf FALSE zurückgesetzt wird.

Ein Reset-Ausgang wird auf FALSE gesetzt, wenn der Ausgang des Elementes auf TRUE gesetzt wird. Der Reset-Ausgang behält diesen Wert, selbst wenn der Ausgang des Elements wieder auf FALSE zurückgesetzt wird.

Wenn Sie den Befehl mehrmals ausführen, wechselt der Ausgang zwischen Set-Ausgang, Reset-Ausgang und normalem Ausgang.

[Extras] / [Ansicht] im FUP

Mit dem Menübefehl [Extras] / [Ansicht] ([Extras] / [View]) wählen Sie die Darstellung eines Baustein aus, der im Funktionsplan-Editor erstellt wurde. Sie können zwischen der Darstellung im Kontaktplan-Editor und der Darstellung im Funktionsplan-Editor auswäh-

len. Dies ist sowohl im Offline- als auch im Online-Modus möglich.

Zoom zum aufgerufenen Baustein

Tastaturbefehl: <Alt>+<Eingabe>

Mit dem Menübefehl [Zoom] öffnen Sie einen aufgerufenen Baustein in seinem Editorfenster.

Dieser Befehl steht im Kontextmenü (<F2>) oder im Menü [Extras] zur Verfügung,

- wenn in Texteditoren der Cursor auf dem Namen des aufgerufenen Bausteins steht
- wenn in grafischen Editoren die Box des Bausteins markiert ist.

Wenn es sich um einen Baustein aus einer Bibliothek handelt, wird der Bibliotheksverwalter aufgerufen und der entsprechende Baustein angezeigt.

Ausschneiden, Kopieren, Einfügen und Löschen in FUP

Folgende Menübefehle befinden sich unter dem Menüpunkt [Bearbeiten] ([Edit]):

- [Ausschneiden] ([Cut])<Strg>+<X>
- [Kopieren] ([Copy]) <Strg>+<C>
- [Einfügen] ([Paste]) <Strg>+<V>
- [Löschen] ([Delete]) <Entf>

Die Menübefehle wirken folgendermaßen:

- Wenn ein Linienkreuz markiert ist, werden die darunterliegenden Zuweisungen, Sprünge oder `Return`-Anweisungen ausgeschnitten, gelöscht oder kopiert.
- Wenn ein Baustein markiert ist, wird der markierte Baustein selbst, sowie alle an den Eingängen anliegenden Verbindungen mit Ausnahme der ersten (obersten) Verbindung ausgeschnitten, gelöscht oder kopiert.
- In allen anderen Fällen wird der gesamte vor der Cursorposition liegende Ast ausgeschnitten, gelöscht oder kopiert.

Nach dem Kopieren oder Ausschneiden liegt der gelöschte bzw. kopierte Teil in der Zwischenablage und kann nun beliebig oft eingefügt werden.

Dazu müssen Sie zunächst die Einfügeposition auswählen. Gültige Einfügepositionen sind Eingänge und Ausgänge.

Wenn ein Baustein in der Zwischenablage liegt, wird der erste Eingang mit dem Ast vor der Einfügeposition verbunden.

Andernfalls wird der gesamte vor der Einfügeposition liegende Ast durch den Inhalt der Zwischenablage ersetzt.

In jedem Fall wird das letzte eingefügte Element mit dem rechts von der Einfügeposition liegenden Ast verbunden.

Hinweis:

Folgendes Problem können Sie durch Ausschneiden und Einfügen lösen:

In der Mitte eines Netzwerks fügen Sie einen neuen Operator ein. Der rechts vom Operator liegende Ast ist nun mit dem ersten Eingang verbunden, soll aber mit dem zweiten Eingang verbunden sein.

Wählen Sie den ersten Eingang aus und führen Sie den Menübefehl [Bearbeiten] / [Ausschneiden] ([Edit] / [Cut]) aus. Anschließend wählen Sie den zweiten Eingang aus und führen den Befehl [Bearbeiten] / [Einfügen] ([Edit] / [Paste]) aus.

Somit hängt der Ast nun am zweiten Eingang.

Der Funktionsplan im Online-Modus

Im Funktionsplan können Sie **Breakpoints** nur auf Netzwerke setzen. Wenn ein Breakpoint auf ein Netzwerk gesetzt wurde, dann wird das Netzwerknummernfeld blau dargestellt. Die Bearbeitung des Programms stoppt dann vor dem Netzwerk, auf dem der Breakpoint steht. In diesem Fall wird das Netzwerknummernfeld rot dargestellt. Beim Steppen (Einzelschritt) springt das Programm von Netzwerk zu Netzwerk.

Der aktuelle Wert jeder Variablen wird dargestellt.

Ausnahme: Wenn der Eingang eines Funktionsblocks ein Ausdruck ist, wird nur die erste Variable des Ausdrucks überwacht (Monitoring).

Mit einem Doppelklick auf eine Variable öffnen Sie den Dialog zum Schreiben einer Variablen. Im Dialog können Sie den aktuellen Wert der Variablen ändern. Bei booleschen Variablen erscheint kein Dialog, ihr Wert wird direkt umgeschaltet (Toggeln).

Der neue Wert wird rot dargestellt und bleibt unverändert. Wenn Sie den Menübefehl [Online] / [Werte schreiben] ([Online] / [Write Value]) ausführen, werden alle Variablen auf die gewählten Werte gesetzt und wieder schwarz dargestellt.

Die Ablaufkontrolle starten Sie mit dem Menübefehl [Online] / [Ablaufkontrolle] ([Online] / [Display Flow Control]). Mit Hilfe der Ablaufkontrolle können Sie die aktuellen Werte ansehen, die in den Netzwerken über die Verbindungslinien übertragen werden.

- Wenn die Verbindungslinien keine booleschen Werte übertragen, wird der Wert in einem separat eingefügten Feld angezeigt.
- Wenn die Verbindungslinien boolesche Werte übertragen, dann werden die Verbindungslinien blau dargestellt, wenn sie TRUE transportieren.

So können Sie den Informationsfluss während des Steuerungsablaufs mit verfolgen.

Wenn Sie den Mauszeiger eine kurze Zeit über einer Variablen halten, wird der Typ, die Adresse und der Kommentar der Variablen in einem Tooltip angezeigt.

5.5.9 Der Kontaktplan-Editor KOP (LD)

Überblick

Die Editoren für die Bausteine bestehen aus einem Deklarationsteil und einem Rumpf. Die Bildschirmdarstellung der Editoren ist durch einen Bildschirmteiler geteilt.

Der KOP-Editor ist ein grafischer Editor.

Die wichtigsten Befehle finden Sie im Kontextmenü, das Sie durch einen Klick mit der rechten Maustaste öffnen.

Nähere Informationen über die Elemente finden Sie auf Seite 41 (→ Kap. 2.4.7, Kontaktplan KOP (LD)).

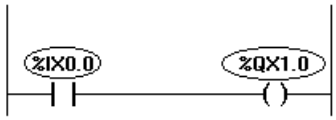
Weitere Informationen finden Sie in den folgenden Abschnitten:

- (→ Abschnitt. Cursorpositionen im KOP-Editor), Seite 237
- (→ Abschnitt. [Einfügen] / [Kontakt] im KOP), Seite 239
- (→ Abschnitt. [Einfügen] / [Paralleler Kontakt] im KOP), Seite 239
- (→ Abschnitt. [Einfügen] / [Funktionsblock] im KOP), Seite 240
- (→ Abschnitt. [Einfügen] / [Spule] im KOP), Seite 240
- (→ Abschnitt. Bausteine mit EN-Eingängen), Seite 240
- (→ Abschnitt. [Einfügen] / [Baustein mit EN] im KOP), Seite 241
- (→ Abschnitt. [Einfügen] / [Einfügen an Baustein] im KOP), Seite 241
- (→ Abschnitt. [Einfügen] / [Sprung] im KOP), Seite 241
- (→ Abschnitt. [Einfügen] / [Return] im KOP), Seite 242
- (→ Abschnitt. [Extras] / [Dahinter Einfügen] im KOP), Seite 242
- (→ Abschnitt. [Extras] / [Darunter Einfügen] im KOP), Seite 242
- (→ Abschnitt. [Extras] / [Darüber Einfügen] im KOP), Seite 242
- (→ Abschnitt. [Extras] / [Negation] im KOP), Seite 242
- (→ Abschnitt. [Extras] / [Set/Reset] im KOP), Seite 242
- (→ Abschnitt. [Extras] / [Zoom] ([Extras] / [Zoom])), Seite 197
- (→ Kap. [Extras] / [Instanz öffnen]), Seite 197
- (→ Kap. Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]), Seite 224
- (→ Abschnitt. Der Kontaktplan im Online-Modus), Seite 243

Cursorpositionen im KOP-Editor

Im KOP-Editor sind folgende Cursorpositionen möglich (schwarz umrahmt bzw. gestrichelt):

1. Textfelder:



20434AXX

2. Kontakte oder Funktionsblöcke:

20435AXX

3. Spulen:

20436AXX

4. Verbindungslinien zwischen Kontakten und Spulen:

20437AXX

Funktionsblock- und Programmaufrufe können wie Kontakte behandelt werden.

Bausteine mit EN-Eingängen und daran geknüpfte andere Bausteine werden wie Bausteine im Funktionsplan behandelt.

Weitere Information über das Editieren dieser Netzwerkteile finden Sie auf Seite 41 (→ Kap. 2.4.7, Kontaktplan KOP (LD)).

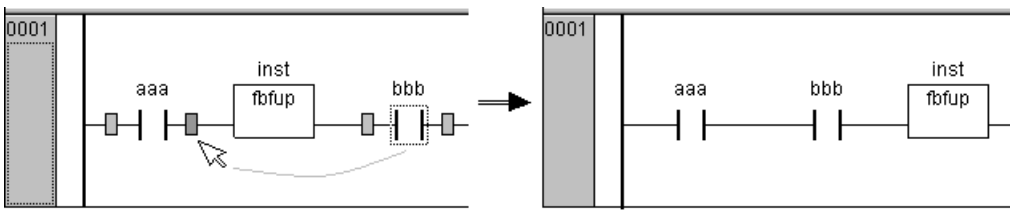
Elemente und Namen im KOP-Editor verschieben

Mit Hilfe von "Drag & Drop" können Sie folgende Komponenten eines Bausteins an eine andere Position innerhalb des Bausteins verschieben:

- ein ganzes Element (Kontakt, Spule, Funktionsblock) eines KOP-Bausteins
- oder nur den Namen (Variablenname, Adresse, Kommentar) eines Elements

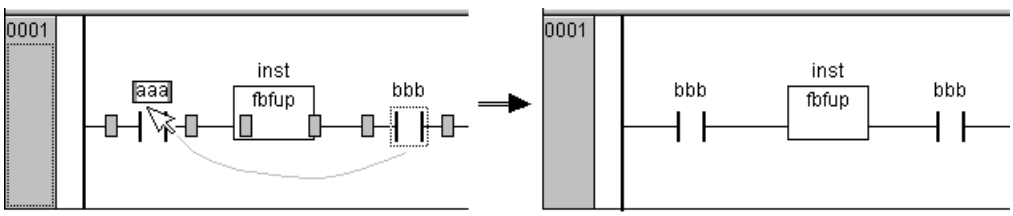
Markieren Sie dazu den gewünschten Kontakt, die Spule oder den Funktionsblock und ziehen Sie ihn bei gedrückter linker Maustaste von der momentanen Position weg. Daraufhin werden alle möglichen Positionen innerhalb der Netzwerke des Bausteins, zu denen Sie das Element verschieben können, durch grau gefüllte Rechtecke angezeigt.

Wenn Sie das Element auf eine dieser Markierungen ziehen, wird diese grün dargestellt. Wenn Sie dann die Maustaste loslassen, wird das Element an der neuen Position eingefügt.



20505AXX

Wenn Sie das Element jedoch auf die Beschriftung (den Variablennamen) eines anderen Elements ziehen, wird der Hintergrund des Variablennamens grün dargestellt. Wenn Sie die Maustaste loslassen, wird der bisherige Name durch den herangezogenen Namen ersetzt. Falls zusätzlich die Adresse und der Kommentar angezeigt werden, bezieht sich das Kopieren auch auf diese Adresse.



20506AXX

[Einfügen] / [Kontakt] im KOP

Schaltfläche: []

Tastaturbefehl: <Strg>+<K>

Mit dem Menübefehl [Einfügen] / [Kontakt] ([Insert] / [Contact]) fügen Sie einen Kontakt vor der markierten Stelle im Netzwerk ein.

Wenn die markierte Stelle eine Spule, oder die Verbindungslinie zwischen den Kontakten und den Spulen ist, wird der neue Kontakt seriell zur bisherigen Kontaktschaltung geschaltet.

Der Kontakt ist mit Standardtext vorbelegt. Sie können diesen Text anklicken und durch die gewünschte Variable bzw. die gewünschte Konstante ersetzen. Dazu können Sie auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen. Anstelle des Variablennamens können Sie eine Adresse eingeben, wenn dies im Dialog [Extras] / [Optionen] ([Extras] / [Options]) so eingestellt ist.

Wenn Sie die Option [Kommentare pro Kontakt] ([Comments per contact]) im Dialog [Extras] / [Optionen] ([Extras] / [Options]) aktiviert haben, können Sie dort die Anzahl der Zeilen für den Variablenkommentar und die Anzahl der Zeilen für den Variablennamen einstellen. Dies ist besonders bei langen Variablennamen nützlich, um das Netzwerk horizontal kompakt zu halten.

Wenn Sie im Dialog [Extras] / [Optionen] ([Extras] / [Options]) die Option [Kommentare pro Kontakt] ([Comments per Contact]) aktiviert ist, können Sie im selben Dialog zusätzlich folgende Angaben machen:

- Anzahl der Zeilen für den Variablenkommentar im Feld [Zeilen für den Variablenkommentar] ([Lines for variable comment])
- Anzahl der Zeilen für den Variablentext im Feld [Zeilen für den Variablentext] ([Lines for variable text]). Das ist besonders bei langen Variablentexten nützlich, um das Netzwerk horizontal kompakt zu halten.

Beachten Sie außerdem die Option [Netzwerke mit Umbrüchen] ([Network with linebreaks]), die Sie ebenfalls im Dialog [Extras] / [Optionen] ([Extras] / [Options]) aktivieren können.

[Einfügen] / [Paralleler Kontakt] im KOP

Schaltfläche: []

Tastaturbefehl: <Strg>+<R>

Mit dem Menübefehl [Einfügen] / [Paralleler Kontakt] ([Insert] / [Parallel Contact]) fügen Sie einen Kontakt parallel zur markierten Stelle im Netzwerk ein.

Der neue Kontakt wird parallel zur gesamten bisherigen Kontaktschaltung geschaltet,

- wenn die markierte Stelle eine Spule ist
- oder wenn die markierte Stelle die Verbindung zwischen den Kontakten und den Spulen ist.

Der Kontakt erhält den Text "???". Sie können diesen Text anklicken und durch die gewünschte Variable bzw. die gewünschte Konstante ersetzen. Dazu können Sie die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen.

Weitere Information über die Eingabe von Adressen, der mehrzeiligen Darstellung des Variablennamens bzw. des Kommentars für den Kontakt finden Sie auf Seite 224 (→ Kap. 5.5.5, Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]).

[Einfügen] / [Funktionsblock] im KOP

Tastaturbefehl: <Strg>+<B>

Mit dem Menübefehl [Einfügen] / [Funktionsblock] ([Insert] / [Function Block]) fügen Sie einen Funktionsblock oder ein Programm als Baustein ein. Vor dem Einfügen markieren Sie die Verbindung zwischen den Kontakten und den Spulen oder die Spule. Wenn sie den Befehl ausführen wird der Eingabehilfe-Dialog geöffnet. Dort wählen Sie einen der zur Verfügung stehenden Standard- und selbst definierten Bausteinen aus.

Der erste Eingang des neuen Bausteins wird auf die Eingangsverbindung gelegt. Der erste Ausgang wird auf die Ausgangsverbindung gelegt. Deshalb müssen diese Variablen ausschließlich als Typ BOOL deklariert sein. Alle anderen Ein- und Ausgänge des Bausteins sind mit dem Text "???" belegt. Diese Vorbelegungen können Sie durch andere Konstanten, Variablen oder Adressen ersetzen. Dazu können Sie auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen.

Weitere Informationen über die Eingabe von Adressen und die mehrzeilige Darstellung des Variablennamens bzw. des Kommentars eines Funktionsblocks finden Sie auf Seite 224 (→ Kap. 5.5.5, Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]).

[Einfügen] / [Spule] im KOP

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<L>

Mit dem Menübefehl [Einfügen] / [Spule] ([Insert] / [Coil]) fügen Sie eine Spule parallel zu den bisherigen Spulen ein.

- Wenn die markierte Stelle die Verbindung zwischen den Kontakten und den Spulen ist, wird die neue Spule als letzte Spule eingefügt.
- Wenn die markierte Stelle eine Spule ist, wird die neue Spule direkt darüber eingefügt.

Die Spule erhält den Text "???". Sie können diesen Text anklicken und durch die gewünschte Variable ersetzen. Dazu können Sie auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen.

Weitere Informationen über die Eingabe von Adressen und die mehrzeilige Darstellung des Variablennamens bzw. des Kommentars einer Spule finden Sie auf Seite 224 (→ Kap. 5.5.5, Netzwerkkommentare, Umbrüche, [Extras] / [Optionen]).

Bausteine mit EN-Eingängen

Wenn das KOP-Netzwerk andere Bausteine aufrufen soll, dann müssen Sie einen Baustein mit einem EN-Eingang einfügen. Diesen Baustein schalten Sie parallel zu den Spulen.

Entwickeln Sie das Netzwerk ausgehend von diesem Baustein weiter. Die Befehle zum Einfügen an einen EN-Baustein finden Sie auf Seite 241 (→ Kap. 6.2.3, [Einfügen] / [Einfügen an Baustein] im KOP).

Ein Operator, ein Funktionsblock, ein Programm oder eine Funktion mit EN-Eingang verhält sich ähnlich wie der entsprechende Baustein im Funktionsplan. Die Ausführung dieser Elemente wird jedoch über den EN-Eingang gesteuert. Der EN-Eingang wird an der Verbindungslinie zwischen Spule und Kontakt angeschlossen. Wenn diese Verbindung auf TRUE gesetzt ist, wird der Baustein ausgewertet.

Wenn Sie einen Baustein mit EN-Eingang angelegt haben, können Sie mit diesem Baustein ein Netzwerk wie im Funktionsplan (FUP) anlegen. In einen EN-Baustein können

Daten von üblichen Operatoren, Funktionen, Funktionsblöcken fließen. Ein EN-Baustein kann Daten an die üblichen Bausteine übertragen.

Wenn Sie im KOP-Editor ein Netzwerk wie im Funktionsplan-Editor (FUP) programmieren möchten, müssen Sie zuerst einen EN-Operator in ein neues Netzwerk einfügen. Anschließend können Sie von diesem Baustein ausgehend ihr Netzwerk weiterbilden wie im FUP-Editor. Das so programmierte Netzwerk verhält sich wie das entsprechende Netzwerk im Funktonsplan (FUP).

[Einfügen] / [Baustein mit EN] im KOP

Mit dem Menübefehl [Einfügen] / [Baustein mit EN] ([Insert] / [Box with EN]) fügen Sie einen Funktionsblock, einen Operator, eine Funktion oder ein Programm mit EN-Eingang in ein Netzwerk ein.

Die markierte Stelle muss die Verbindung zwischen Kontakt und Spule oder eine Spule sein. Der neue Baustein wird parallel unterhalb von den Spulen eingefügt und trägt zunächst die Bezeichnung `AND`. Diese Bezeichnung können Sie wunschgemäß ändern. Dazu können Sie auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen. Zur Verfügung stehen Standard- und selbst definierte Bausteine.

[Einfügen] / [Einfügen an Baustein] im KOP

Mit dem Menübefehl [Einfügen] / [Einfügen an Baustein] ([Insert] / [Insert at Blocks]) fügen Sie an einen bereits eingefügten Baustein (auch ein Baustein mit EN-Eingang), weitere Elemente ein. Die Befehle dieses Menüpunkts sind an denselben Cursorpositionen ausführbar wie die entsprechenden Befehle im Funktionsplan FUP.

- Mit dem Menübefehl [Einfügen] / [Einfügen an Baustein] / [Eingang] ([Insert] / [Insert at Blocks] / [Input]) fügen Sie einen neuen Eingang an den Baustein an.
- Mit dem Menübefehl [Einfügen] / [Einfügen an Baustein] / [Ausgang] ([Insert] / [Insert at Blocks] / [Output]) fügen Sie einen neuen Ausgang an den Baustein an.
- Mit dem Menübefehl [Einfügen] / [Einfügen an Baustein] / [Box] ([Insert] / [Insert at Blocks] / [Box]) fügen Sie einen weiteren Baustein an. Die Vorgehensweise entspricht der Vorgehensweise, die auf Seite 232 (→ Abschnitt. [Einfügen] / [Baustein] im FUP) beschrieben ist.
- Mit dem Menübefehl [Eingfügen] / [Einfügen an Baustein] / [Zuweisung] ([Insert] / [Insert at Blocks] / [Assign]) fügen Sie eine Zuweisung zu einer Variablen ein. Zunächst wird die Zuweisung durch drei Fragezeichen "???" dargestellt, die Sie anklicken und durch die gewünschte Variable ersetzen können. Sie können auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste aufrufen.

[Einfügen] / [Sprung] im KOP

Mit dem Menübefehl [Einfügen] / [Sprung] ([Insert] / [Jump]) fügen Sie einen Sprung am Ende des Kontaktplans parallel zu den bisherigen Spulen ein. Wenn die eingehende Leitung auf TRUE gesetzt ist, wird der Sprung an die bezeichnete Marke durchgeführt.

Bevor Sie einen Sprung einfügen, markieren Sie eine Spule oder die Verbindung zwischen Kontakt und Spule.

Der Sprung erhält automatisch die Bezeichnung "Label". Sie können diesen Text anklicken und durch die gewünschte Sprungmarke ersetzen.

[Einfügen] / [Return] im KOP

Mit dem Menübefehl [Einfügen] / [Return] ([Insert] / [Return]) fügen Sie eine `Return`-Anweisung am Ende parallel zu den bisherigen Spulen ein. Wenn die eingehende Leitung auf "AN" gesetzt ist, wird die Bearbeitung des Bausteins dieses Netzwerks abgebrochen.

Bevor Sie die `Return`-Anweisung einfügen markieren Sie eine Spule oder die Verbindung zwischen Kontakt und Spule.

[Extras] / [Dahinter Einfügen] im KOP

Mit dem Menübefehl [Extras] / [Dahinter Einfügen] ([Extras] / [Paste after]) fügen Sie den Inhalt der Zwischenablage als seriellen Kontakt nach der Markierungsstelle ein.

Diesen Befehl können Sie nur ausführen, wenn der Inhalt der Zwischenablage und die markierte Stelle Netzwerke aus Kontakten sind.

[Extras] / [Darunter Einfügen] im KOP

Tastaturbefehl: <Strg>+<U>

Mit dem Menübefehl [Extras] / [Darunter Einfügen] ([Extras] / [Paste below]) fügen Sie den Inhalt der Zwischenablage als parallelen Kontakt unter der Markierungsstelle ein.

Diesen Menübefehl können Sie nur ausführen, wenn der Inhalt der Zwischenablage und die markierte Stelle Netzwerke aus Kontakten sind.

[Extras] / [Darüber Einfügen] im KOP

Mit dem Menübefehl [Extras] / [Darüber Einfügen] ([Extras] / [Paste above]) fügen Sie den Inhalt der Zwischenablage als parallelen Kontakt über der Markierungsstelle ein.

Diesen Befehl können Sie nur ausführen, wenn der Inhalt der Zwischenablage und die markierte Stelle Netzwerke aus Kontakten sind.

[Extras] / [Negation] im KOP

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<N>

Mit dem Menübefehl [Extras] / [Negation] ([Extras] / [Negate]) negieren Sie – je nach aktueller Cursorposition – folgende Komponenten:

- einen Kontakt
- eine Spule
- eine Sprung-Anweisung
- eine `Return`-Anweisung
- einen Eingang bzw. Ausgang von EN-Bausteinen

Zwischen den runden Klammern der Spule bzw. zwischen den geraden Strichen des Kontakts erscheint ein Schrägstrich (`(/)` bzw. `|/|`).

Bei Sprüngen, Returns, Ein- bzw. Ausgängen von EN-Bausteinen erscheint wie im FUP-Editor ein kleiner Kreis auf der Verbindung.

Die Spule schreibt nun den negierten Wert der Eingangsverbindung in die zugehörige

boolesche Variable. Ein negierter Kontakt schaltet genau dann den Zustand des Eingangs auf den Ausgang, wenn die zugehörige boolesche Variable auf FALSE gesetzt wird.

Wenn ein Sprung oder ein Return markiert ist, wird der Eingang dieses Sprungs bzw. Returns negiert.

Die Negation können Sie durch eine erneute Negation löschen.

[Extras] / [Set/Reset] im KOP

Mit dem Menübefehl [Extras] / [Set/Reset] ([Extras] / [Set/Reset]) wandeln Sie eine Spule eine in eine Set-Spule um. Eine Set-Spule überschreibt den Wert TRUE in der zugehörigen booleschen Variablen niemals. Wenn der Wert dieser Variablen einmal auf TRUE gesetzt wurde, bleibt er immer auf TRUE gesetzt.

Eine Set-Spule wird mit einem [S] im Spulensymbol gekennzeichnet.

Wenn Sie diesen Befehl erneut ausführen, wandeln Sie die Set-Spule in eine Reset-Spule um. Eine Reset-Spule überschreibt den Wert

FALSE in der zugehörigen booleschen Variablen niemals. Wenn der Wert dieser Variablen einmal auf FALSE gesetzt wurde, dann bleibt er immer auf FALSE gesetzt.

Eine Reset-Spule wird mit einem [R] im Spulensymbol gekennzeichnet.

Wenn Sie diesen Befehl mehrmals ausführen, dann wandeln Sie diese Spule zwischen Set-Spule, Reset-Spule und normaler Spule um.

Der Kontaktplan im Online-Modus

Im Online-Modus werden im Kontaktplan alle Kontakte, Spulen und Leitungen, die im Zustand "AN" (*TRUE*) sind, blau dargestellt.

An den Ein- und Ausgängen von Funktionsblöcken werden die Werte der entsprechenden Variablen angezeigt.

Breakpoints können Sie nur auf Netzwerke setzen. Beim Steppen springt das Programm von Netzwerk zu Netzwerk.

Bei aktivierter Ablaufkontrolle [Online] / [Ablaufkontrolle] ([Online] / [Display Flow Control]) werden die Nummernfelder der durchlaufenen Netzwerke grün dargestellt.

Wenn Sie den Mauszeiger eine kurze Zeit über eine Variable halten, wird der Typ, die Adresse und der Kommentar der Variablen in einem Tooltip angezeigt.

5.5.10 Der Ablaufsprachen-Editor AS (SFC)

Übersicht

Alle Editoren für Bausteine bestehen aus einem Deklarationsteil und einem Rumpf. Die Bildschirmdarstellung der Editoren ist durch einen Bildschirmteiler geteilt.

Der Ablaufsprachen-Editor ist ein grafischer Editor.

Die wichtigsten Befehle finden Sie im Kontextmenü, das Sie durch einen Klick mit der rechten Maustaste öffnen.

Tooltips zeigen im Offline-Modus, im Online-Modus und auch im gezoomten Status die vollständigen Namen bzw. Ausdrücke von Schritten, Transitionen, Sprüngen, Sprungmarken, Qualifiern oder assoziierten Aktionen an.

Nähere Informationen über die Sprache finden Sie auf Seite 33 (→ Kap. 2.4.4, Ablaufsprache AS (SFC)).

Der Editor muss für die Besonderheiten der Ablaufsprache geeignet sein.

Weitere Informationen finden Sie in den folgenden Abschnitten:

- (→ Abschnitt. Blöcke markieren im AS), Seite 244
- (→ Abschnitt. [Einfügen] / [Schritt-Transition (davor)] im AS), Seite 245
- (→ Abschnitt. [Einfügen] / [Schritt-Transition (danach)] im AS), Seite 245
- (→ Abschnitt. [Einfügen] / [Alternativzweig (rechts)] im AS), Seite 245
- (→ Abschnitt. [Einfügen] / [Alternativzweig (links)] im AS), Seite 245
- (→ Abschnitt. [Einfügen] / [Parallelzweig (rechts)] im AS), Seite 246
- (→ Abschnitt. [Einfügen] / [Parallelzweig (links)] im AS), Seite 246
- (→ Abschnitt. [Einfügen] / [Sprung] im AS), Seite 246
- (→ Abschnitt. [Einfügen] / [Transition-Sprung] im AS), Seite 246
- (→ Abschnitt. [Einfügen] / [Eingangsaktion hinzufügen] im AS), Seite 247
- (→ Abschnitt. [Einfügen] / [Ausgangsaktion hinzufügen] im AS), Seite 247
- (→ Abschnitt. [Extras] / [Parallelzweig einfügen (rechts)] im AS), Seite 247
- (→ Abschnitt. [Extras] / [Marke zu Parallelzweig hinzufügen] im AS), Seite 247
- (→ Abschnitt. Sprungmarke löschen im AS), Seite 248
- (→ Abschnitt. [Extras] / [Einfügen danach] im AS), Seite 248
- (→ Abschnitt. [Extras] / [Zoom Aktion/Transition] im AS), Seite 248
- (→ Abschnitt. [Extras] / [Lösche Aktion/Transition] im AS), Seite 249
- (→ Abschnitt. [Extras] / [Schritte Attribute] im AS), Seite 249
- (→ Abschnitt. [Extras] / [Zeitenüberblick] im AS), Seite 250
- (→ Abschnitt. [Extras] / [Optionen] im AS), Seite 251
- (→ Abschnitt. [Extras] / [Aktion assoziieren] im AS), Seite 251
- (→ Abschnitt. [Extras] / [IEC-Schritte benutzen] im AS), Seite 252
- (→ Abschnitt. Die Ablaufsprache im Online-Modus), Seite 252

Blöcke markieren im AS

Ein markierter Block ist eine Menge von AS-Elementen, die von einem gepunkteten Rechteck umgeben sind.

Ein Element (einen Schritt, eine Transition oder einen Sprung) markieren Sie mit den Pfeiltasten oder indem Sie mit der linken Maustaste auf das Element klicken.

Eine Menge von mehreren Elementen markieren Sie, indem Sie zusätzlich zu einem be-

reits markierten Block die <Umschalt>-Taste drücken und das Element in der linken oder rechten unteren Ecke der Menge auswählen. Die Auswahl, die Sie so markieren, ist die kleinste zusammenhängende Menge von Elementen, die diese beiden Elemente beinhaltet.

Hinweis:

Einen Schritt können Sie nur zusammen mit der vorangehenden oder nachfolgenden Transition löschen!

[Einfügen] / [Schritt-Transition (davor)] im AS

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<T>

Mit dem Menübefehl [Einfügen] / [Schritt-Transition (davor)] ([Insert] / [Step-Transition (before)]) fügen Sie einen Schritt und eine Transition vor der markierten Position ein.

[Einfügen] / [Schritt-Transition (danach)] im AS

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<E>

Mit dem Menübefehl [Einfügen] / [Schritt-Transition (danach)] ([Insert] / [Step-Transition (after)]) fügen Sie einen Schritt und eine Transition nach der ersten Transition der markierten Position ein.

[Bearbeiten] / [Löschen] ([Edit] / [Delete]) im AS

Tastaturbefehl: <Entf>

Sie können einen Schritt nur zusammen mit der vorangehenden oder nachfolgenden Transition löschen. Markieren Sie dazu den Schritt und die Transition und führen Sie den Befehl [Bearbeiten] / [Löschen] ([Edit] / [Delete]) aus oder drücken Sie die <Entf>-Taste.

[Einfügen] / [Alternativzweig (rechts)] im AS

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<A>

Mit dem Menübefehl [Einfügen] / [Alternativzweig (rechts)] ([Insert] / [Alternative Branch (right)]) fügen Sie eine Alternativ-Verzweigung als rechte Verzweigung des markierten Blocks ein.

Der markierte Block muss mit einer Transition beginnen und enden.

Der neue Zweig besteht dann aus einer Transition.

[Einfügen] / [Alternativzweig (links)] im AS

Schaltfläche: [ ]

Mit dem Menübefehl [Einfügen] / [Alternativzweig (links)] ([Insert] / [Alternative Branch (left)]) fügen Sie eine Alternativ-Verzweigung als linke Verzweigung des markierten

Blocks ein.

Der markierte Block muss mit einer Transition beginnen und enden.

Der neue Zweig besteht dann aus einer Transition.

[Einfügen] / [Parallelzweig (rechts)] im AS

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<L>

Mit dem Menübefehl [Einfügen] / [Parallelzweig (rechts)] ([Insert] / [Parallel Branch (right)]) fügen Sie eine parallele Verzweigung als rechte Verzweigung des markierten Blocks ein.

Der markierte Block muss mit einem Schritt beginnen und enden.

Der neue Zweig besteht dann aus einem Schritt.

Um Sprünge auf die entstandene Parallelverzweigung zu ermöglichen, müssen Sie an der Parallelverzweigung eine Sprungmarke einfügen.

[Einfügen] / [Parallelzweig (links)] im AS

Schaltfläche: [ ]

Mit dem Menübefehl [Einfügen] / [Parallelzweig (links)] ([Insert] / [Parallel Branch (left)]) fügen Sie eine parallele Verzweigung als linke Verzweigung des markierten Blocks ein.

Der markierte Block muss mit einem Schritt beginnen und enden.

Der neue Zweig besteht dann aus einem Schritt.

Um Sprünge auf die entstandene Parallelverzweigung zu ermöglichen, müssen Sie an der Parallelverzweigung eine Sprungmarke einfügen.

[Einfügen] / [Sprung] im AS

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<U>

Mit dem Menübefehl [Einfügen] / [Sprung] ([Insert] / [Jump]) fügen Sie am Ende des Zweigs eine Sprung-Answeisung ein.

Die Verzweigung muss eine Alternativ-Verzweigung sein.

Nachdem Sie den Sprung eingefügt haben, klicken Sie auf den Text "Step" und ersetzen diesen durch den Schrittnamen bzw. die Sprungmarke der Parallelverzweigungen, zu dem/der das Programm springen soll.

[Einfügen] / [Transition-Sprung] im AS

Schaltfläche: [ ]

Mit dem Menübefehl [Einfügen] / [Transition-Sprung] ([Insert] / [Transition-Jump]) fügen Sie eine Transition und einen Sprung am Ende der markierten Verzweigung ein.

Die markierte Verzweigung muss eine parallele Verzweigung sein.

Nachdem Sie die Transition und den Sprung eingefügt haben, klicken Sie auf den Text "Step" und ersetzen diesen durch den Schrittnamen bzw. die Sprungmarke der Parallelverzweigung, zu dem/der das Programm springen soll.

[Einfügen] / [Eingangsaktion hinzufügen] im AS

Mit dem Menübefehl [Einfügen] / [Eingangsaktion hinzufügen] ([Insert] / [Add Entry-Action]) fügen Sie zu einem Schritt eine Eingangsaktion hinzu.

Eine Eingangsaktion wird nur einmal ausgeführt, gleich nachdem der Schritt aktiv geworden ist. Die Eingangsaktion kann in einer beliebigen Sprache implementiert werden.

Ein Schritt mit Eingangsaktion wird durch das Symbol [E] in der linken unteren Ecke des Schritts gekennzeichnet.

[Einfügen] / [Ausgangsaktion hinzufügen] im AS

Mit dem Menübefehl [Einfügen] / [Ausgangsaktion hinzufügen] ([Insert] / [Add Exit-Action]) fügen Sie zu einem Schritt eine Ausgangsaktion hinzu.

Eine Ausgangsaktion wird nur einmal ausgeführt, bevor der Schritt deaktiviert wird. Die Ausgangsaktion kann in einer beliebigen Sprache implementiert werden.

Ein Schritt mit Ausgangsaktion wird durch das Symbol [X] in der rechten unteren Ecke des Schritts gekennzeichnet.

[Extras] / [Parallelzweig einfügen (rechts)] im AS

Mit dem Menübefehl [Extras] / [Parallelzweig einfügen (rechts)] ([Extras] / [Paste Parallel Branch (right)]) fügen Sie den Inhalt der Zwischenablage als rechte Parallelverzweigung des markierten Blocks ein.

Der markierte Block muss mit einem Schritt beginnen und enden.

Der Inhalt der Zwischenablage muss ebenfalls ein AS-Block sein, der mit einem Schritt beginnt und endet.

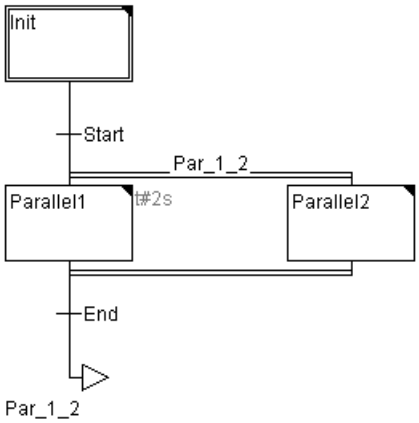
[Extras] / [Marke zu Parallelzweig hinzufügen] im AS

Mit dem Menübefehl [Extras] / [Marke zu Parallelzweig hinzufügen] ([Extras] / [Add Label to Parallel Branch]) fügen Sie zu einer Parallelverzweigung eine Sprungmarke hinzu.

Um in die neu eingefügte Parallelverzweigung eine Sprungmarke einzufügen, klicken Sie auf die vor der Parallelverzweigung liegende Transition und führen den Befehl [Extras] / [Marke zu Parallelzweig hinzufügen] ([Extras] / [Add Label to Parallel Branch]) aus. Die Parallelverzweigung hat nun den Standardnamen "Parallel" und eine angehängte laufende Nummer, die Sie nach den Regeln für Bezeichnernamen verändern können.

Beispiel

Im folgenden Beispiel wurde der Standardname "Parallel" durch "Par_1_2" ersetzt. Der Sprung nach der Transition "End" wurde auf diese Sprungmarke gelenkt.



20438AXX

Sprungmarke löschen im AS

Eine Sprungmarke löschen Sie, indem Sie den Text der Sprungmarke löschen.

[Extras] / [Einfügen danach] im AS

Mit dem Menübefehl [Extras] / [Einfügen danach] ([Extras] / [Paste after]) fügen Sie den AS-Block in der Zwischenablage nach dem ersten Schritt bzw. der ersten Transition des markierten Blocks ein.

Das normale Kopieren mit dem Menübefehl [Bearbeiten] / [Einfügen] ([Edit] / [Paste]) fügt den AS-Block vor dem markierten Block ein.

Der Menübefehl [Extras] / [Einfügen danach] ([Extras] / [Paste after]) wird nur ausgeführt, wenn die resultierende AS-Struktur gemäß den Sprachnormen korrekt ist.

[Extras] / [Zoom Aktion/Transition] im AS

Tastaturbefehl: <Alt>+<Eingabe>

Mit dem Menübefehl ([Extras] / [Zoom Aktion/Transition] ([Extras] / [Zoom Action/Transition]) fügen Sie zu einem Block eine Aktion bzw. Transition hinzu.

Die markierte Block muss ein Schritt oder eine Transition sein.

Die Aktion des ersten Schritts eines markierten Blocks bzw. der Rumpf der ersten Transition eines markierten Blocks werden in der Programmiersprache geladen, in der der Block geschrieben ist.

Wenn die Aktion oder der Rumpf der Transition leer ist, dann wählen Sie im Dialog die Sprache der Aktion bzw. Transition aus.

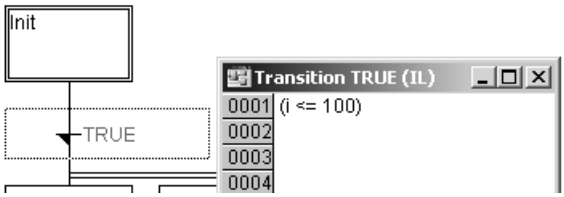
.

Hinweis:

Beachten Sie bei Transitionen, dass die im Editor geschriebene Bedingung hat Vorrang vor der Bedingung, die direkt an der Transitionsmarke steht!

Beispiel

Wenn im Editor `i>100` steht, dann gilt für die Transitionsbedingung FALSE, obwohl an der Transitionsmarke TRUE steht!



20508AEN

[Extras] / [Lösche Aktion/Transition] im AS

Mit dem Menübefehl [Extras] / [Lösche Aktion/Transition] ([Extras] / [Clear Action/Transition]) löschen Sie die Aktionen des ersten Schritts des markierten Blocks bzw. die erste Transition des markierten Blocks.

In Abhängigkeit von der Cursorposition hat der Menübefehl unterschiedliche Auswirkungen:

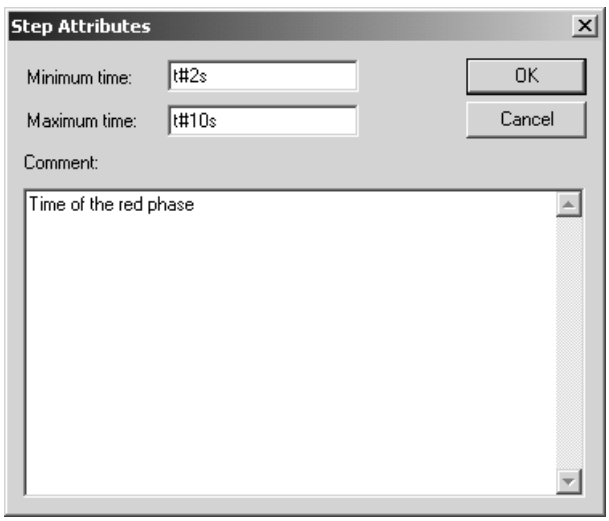
- Wenn an einem Schritt nur die Aktion, die Eingangsaktion oder die Ausgangsaktion implementiert ist, wird diese gelöscht.

  Andernfalls erscheint ein Dialog, in dem Sie auswählen, welche Aktion bzw. Aktionen gelöscht werden.
- Wenn der Cursor in einer Aktion eines IEC-Schritts steht, wird nur diese Assoziation gelöscht.
- Wenn ein IEC-Schritt mit einer assoziierten Aktion markiert ist, wird diese Assoziation gelöscht.

  Bei einem IEC-Schritt mit mehreren Aktionen erscheint ein Dialog zur Auswahl der Aktion, die gelöscht werden soll.

[Extras] / [Schritte Attribute] im AS

Mit dem Menübefehl [Extras] / [Schritte Attribute] ([Extras] / [Steps Attributes]) öffnen Sie den Dialog, in dem Sie die Attribute des markierten Schritts eingeben können.



20409AEN

Folgende Attribute können Sie einstellen:

[Minimale Zeit]	[Minimum time]	Geben Sie die Zeitdauer ein, die die Bearbeitung dieses Schritts mindestens dauern soll.
[Maximale Zeit]	[Maximum time]	Geben Sie die Zeitdauer ein, die die Bearbeitung des Schrittes höchstens dauern darf. Beachten Sie, dass die Einträge vom Typ `TIME` sind. Verwenden Sie deshalb eine `TIME`-Konstante (z. B. `T#3s`) oder eine Variable vom Typ `TIME`.
[Kommentar]	[Comment]	Bei Bedarf geben Sie einen Kommentar zum Schritt ein. Sie können einstellen, ob im AS-Editor die Kommentare oder die Zeiteinstellung zu den Schritten dargestellt werden. Die Einstellung nehmen Sie über den Menübefehl [Extras] / [Optionen] ([Extras] / [Options]) im Dialog [Ablaufsprachen Optionen] ([Sequential Function Chart Options]) vor. Rechts neben dem Schritt erscheint dann entweder der Kommentar oder die Zeiteinstellungen.

Bei Überschreiten der Maximalzeit werden AS-Flags gesetzt, die der Benutzer abfragen kann.

Beispiel

Im folgenden Beispiel ist ein Schritt dargestellt, dessen Ausführung mindestens zwei und höchstens zehn Sekunden dauern soll. Im Online-Modus wird zusätzlich zu diesen beiden Zeitgrenzen angezeigt, wie lange der Schritt bereits aktiv ist.

20439AXX

[Extras] / [Zeitenüberblick] im AS

Mit dem Menübefehl [Extras] / [Zeitenüberblick] ([Extras] / [Time Overview]) öffnen Sie den Dialog [AS Zeitgrenzenübersicht] ([SFC Time Limit Overview]), in dem Sie die Zeiteinstellungen der AS-Schritte einstellen können:

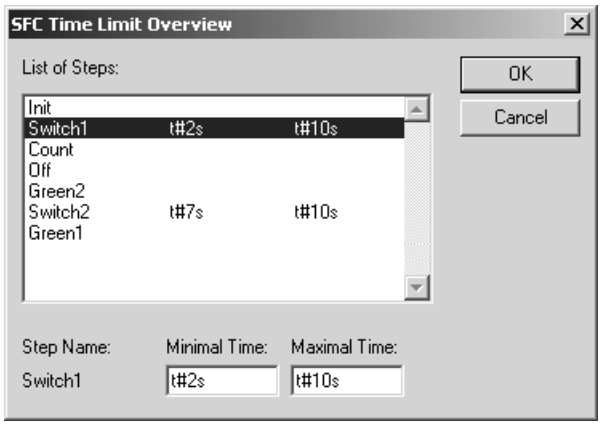
Im Dialog [AS Zeitgrenzenübersicht] werden alle Schritte des AS-Bausteins dargestellt. Wenn Sie zu einem Schritt eine Zeitbegrenzung angegeben haben, dann wird diese rechts vom Schritt angezeigt (zuerst die Untergrenze, dann die Obergrenze).

Die Zeitbegrenzungen können Sie ändern, indem Sie in der Übersicht auf den gewünschten Schritt klicken und in den unteren Feldern die minimale Zeit oder maximale Zeit eingeben. Beachten Sie, dass die Einträge vom Typ `TIME` sind, verwenden Sie deshalb eine `TIME`-Konstante (z. B. `T#3s`) oder eine Variable vom Typ `TIME`. Um die Änderungen zu übernehmen, schließen Sie den Dialog mit [OK].

Beispiel

In diesem Beispiel wurde für die Schritte `Switch1` und `Switch2` eine Zeitbegrenzung eingestellt. Der Schritt `Switch1` dauert mindestens zwei und höchstens zehn Sekun-

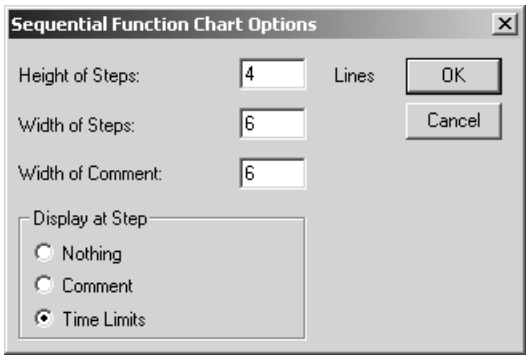
den. Der Schritt `Switch2` dauert mindestens sieben und höchstens acht Sekunden.



20410AEN

[Extras] / [Optionen] im AS

Mit dem Menübefehl [Extras] / [Optionen] ([Extras] / [Options]) öffnen Sie den Dialog [Ablaufsprache Optionen] ([Sequential Functions Chart Options]).



20411AEN

Folgende Optionen können Sie in dem Dialog einstellen:

[Schritthöhe]	[Height of Steps]	Geben Sie ein, wie viele Zeilen ein AS-Schritt in Ihrem AS-Editor hoch sein soll. (Standardeinstellung: 4)
[Schrittbreite]	[Width of Steps]	Geben Sie ein, wie viele Spalten ein AS-Schritt in Ihrem AS-Editor breit sein soll . (Standardeinstellung: 6)
[Kommentar-breite]	[Width of Com-ment]	Geben Sie ein, wie viele Spalten dargestellt werden, wenn Sie den Kommentar beim Schritt mit anzeigen lassen.
[Anzeige beim Schritt]	[Display at Step]	Wählen Sie aus, welche der Attribute, die Sie im Dialog [Schritte Attribute] ([Step Attributes]) eingestellt haben, angezeigt werden.

[Extras] / [Aktion assoziieren] im AS

Mit dem Menübefehl [Extras] / [Aktion assoziieren] ([Extras] / [Assiciate Action]) können Aktionen und boolesche Variablen zu IEC-Schritten assoziiert werden.

Rechts neben dem IEC-Schritt wird ein weiteres zweigeteiltes Kästchen für die Assoziation einer Aktion angehängt. Vorbelegt ist das Kästchen mit dem Qualifier `N` und dem Namen `Action`. Diese Vorbelegungen können Sie mit der Eingabehilfe ändern, die Sie mit der <F2>-Taste öffnen.

Einem IEC-Schritt können Sie maximal neun Aktionen zuweisen!

Neue Aktionen für IEC-Schritte fügen Sie mit dem Menübefehl [Projekt] / [Aktion hinzufügen] ([Project] / [Add Action]) hinzu.

[Extras] / [IEC-Schritte benutzen] im AS

Schaltfläche: [ ]

Wenn der Menübefehl [Extras] / [IEC-Schritte benutzen] ([Extras] / [Use IEC-Steps]) aktiviert ist, werden beim Einfügen von Schritt-Transitionen und Parallelzweigen anstelle von einfachen Schritten IEC-Schritte eingefügt.

Wenn diese Option aktiviert ist, steht vor dem Menüpunkt ein Haken und die Schaltfläche ist gedrückt.

Wenn diese Option aktiviert ist, wird beim Einfügen eines AS-Bausteins der Init-Schritt als IEC-Schritt angelegt.

Diese Einstellung wird in der Datei `PLCEditor.ini` gespeichert und beim nächsten Start des PLC-Editors wiederhergestellt.

Die Ablaufsprache im Online-Modus

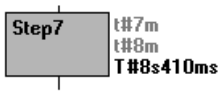
Der Ablaufsprachen-Editor stellt im Online-Modus die momentan aktiven Schritte blau dar.

Bei entsprechender Einstellung im Dialog [Extras] / [Optionen] ([Extras] / [Options]) wird neben den Schritten die Zeitüberwachung dargestellt.

Unter den eingegebenen Unter- und Obergrenzen finden Sie eine dritte Zeitangabe, die anzeigt, wie lange der Schritt bereits aktiv ist.

Beispiel

In der folgenden Abbildung ist der abgebildete Schritt seit acht Sekunden und 410 Millisekunden aktiv. Er muss mindestens sieben Minuten aktiv sein, bevor der Schritt verlassen wird.



20440AXX

Mit dem Menübefehl [Online] / [Breakpoint an/aus] ([Online] / [Toggle Breakpoint]) setzen Sie einen Breakpoint auf einen Schritt. Die Bearbeitung wird dann vor Ausführung dieses Schritts bzw. vor der Programmstelle der Aktion gestoppt. Schritte bzw. Programmstellen, auf die ein Breakpoint gesetzt ist, sind hellblau markiert.

Wenn in einer Parallelverzweigung mehrere Schritte aktiv sind, wird der aktive Schritt, dessen Aktion als nächstes bearbeitet wird, rot dargestellt.

Wenn IEC-Schritte verwendet werden, werden alle aktiven Aktionen im Online-Betrieb blau dargestellt.

Der Ablaufsprachen-Editor unterstützt das schrittweise Steppen:

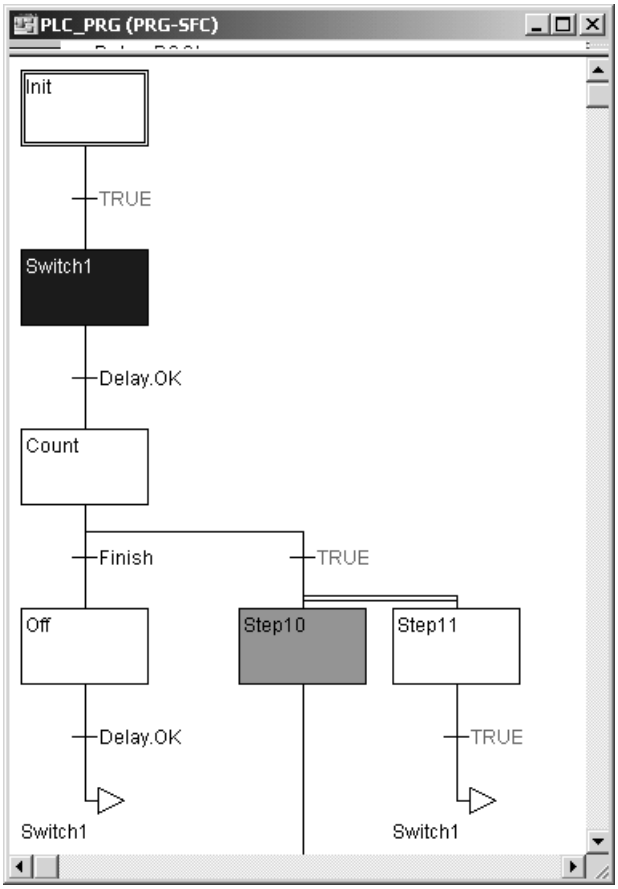
Mit dem Menübefehl [Online] / [Einzelschritt über] ([Online] / [Step over]) springt das Programm zum nächsten Schritt und führt dessen Aktion aus.

In Abhängigkeit von der aktuellen Position im Programm hat der Menübefehl unterschiedliche Auswirkungen:

- Bei einem Schritt in einem linearen Ablauf eines Bausteins oder einem Schritt im äußersten rechten Parallelzweig eines Bausteins wird der AS-Baustein verlassen und zum aufrufenden Element zurückgekehrt. Wenn der Baustein das Hauptprogramm ist, beginnt der nächste Zyklus.
- Bei einem Schritt, der nicht im äußersten rechten einer Parallelverzweigung ist, wird zum aktiven Schritt im nächsten Parallelzweig gesprungen.
- Bei der letzte Breakpoint-Position innerhalb einer AS-Aktion wird zum aufrufenden Element des AS-Programms gesprungen.
- Bei der letzten Breakpoint-Position innerhalb einer IEC-Aktion wird zum aufrufenden Element des AS-Programms gesprungen.
- Bei der letzten Breakpoint-Position innerhalb einer Eingangsaktion/oder Ausgangsaktion wird zum ersten aktiven Schritt gesprungen.

Mit dem Menübefehl [Online] / [Einzelschritt in] ([Online / [Step in]) springt das Programm zusätzlich in die Aktionen hinein. Wenn das Programm in eine Eingangs-, Ausgangs- oder IEC-Aktion springen soll, müssen Sie dort einen Breakpoint setzten. Innerhalb der Aktionen stehen Ihnen alle Debugging-Funktionalitäten des entsprechenden Editors zur Verfügung.

Wenn Sie den Mauszeiger im Deklarationseditor eine kurze Zeit über eine Variablen halten, wird der Typ, die Adresse und der Kommentar der Variablen in einem Tooltip angezeigt.



20412AEN

Hinweis:

Wenn Sie einen Schritt umbenennen und **Online Change** durchführen, während genau dieser Schritt aktiv ist, stoppt das Programm in einem undefinierten Zustand!

Weitere Informationen dazu finden Sie auf Seite 178 (→ Kap. 4.6.2, [Online] / [Einloggen] ([Online] / [Login])).

Abarbeitungsreihenfolge der Elemente einer Schrittkette:

1. Zunächst werden bei IEC-Aktionen alle **Action Control Block-Flags** zurückgesetzt, die in dieser Schrittkette verwendet werden. Eine Ausnahme sind die Flags von IEC-Aktionen, die innerhalb von Aktionen aufgerufen werden.
2. Für alle Schritte wird geprüft, ob die Bedingung für die Ausführung der Ausgangsaktion erfüllt ist. Gegebenenfalls wird diese Ausgangsaktion ausgeführt. Diese Prüfung erfolgt gemäß der Reihenfolge, die durch die Schrittkette festgelegt ist, von oben nach unten und von links nach rechts.
3. Für alle Schritte wird geprüft, ob die Bedingung für die Ausführung der Eingangsaktion erfüllt ist. Gegebenenfalls wird diese Eingangsaktion ausgeführt. Diese Prüfung erfolgt gemäß der Reihenfolge, die durch die Schrittkette festgelegt ist, von oben nach unten und von links nach rechts.
4. Für alle Schritte wird in der Reihenfolge, die sie in der Schrittkette einnehmen, folgende Aktionen ausgeführt:
   - Die abgelaufene Zeit wird gegebenenfalls in die dazugehörige Schrittvariable kopiert.
   - Eine Zeitüberschreitung wird gegebenenfalls überprüft und die AS-Error-Flags entsprechend bedient.

- Wenn kein IEC-Schritt vorliegt, wird nun die dazugehörige Aktion ausgeführt.

5. Die IEC-Aktionen, die in der Schrittkette verwendet werden, werden in alphabetischer Reihenfolge ausgeführt. Dabei durchläuft das Programm die Liste der Aktionen in zwei Durchläufen. Im ersten Durchlauf werden alle im aktuellen Zyklus deaktivierten IEC-Aktionen ausgeführt. Im zweiten Durchlauf werden alle im aktuellen Zyklus aktiven IEC-Aktionen ausgeführt.
6. Die Transitionen werden ausgewertet.

   Der nachfolgende Schritt wird aktiviert, wenn folgende Ereignisse zutreffen:

   - Der nachfolgende Schritt war im aktuellen Zyklus aktiv
   - Die nachfolgende Transition ist auf TRUE gesetzt
   - Die minimal aktive Zeit ist abgelaufen.

Hinweis:

Beachten Sie bei der Implementierung von Aktionen, dass möglicherweise eine Aktion in einem Zyklus mehrfach ausgeführt wird, weil sie in mehreren Schrittketten assoziiert ist.

Beispiel

Ein AS-Baustein kann zwei IEC-Aktionen `A` und `B` besitzen, die beide in AS implementiert sind und die beide die IEC-Aktion `C` aufrufen.

Im selben Zyklus können die IEC-Aktionen `A` und `B` aktiv sein und in beiden IEC-Aktionen kann wiederum die IEC-Aktion `C` aktiv sein. Somit kann die IEC-Aktion `C` zweimal aufgerufen werden.

Wird die selbe IEC-Aktion gleichzeitig in verschiedenen Ebenen eines AS-Programms verwendet, kann dies durch die oben beschriebene Abarbeitungsreihenfolge zu unerwünschten Effekten führen. Deshalb wird in diesem Fall eine Fehlermeldung ausgegeben. Möglicherweise kann dies bei der Bearbeitung von Projekten auftreten, die mit älteren Versionen des PLC-Editors erstellt wurden!

Hinweis:

Beim Überwachen (Monitoring) von Ausdrücken (z. B. `A AND B`) in Transitionen wird nur der "Gesamtwert" der Transition dargestellt.

5.5.11 Der Freigrafische Funktionsplan-Editor CFC

Übersicht

Beim CFC-Editor werden nicht wie im gewöhnlichen FUP-Editor Netzwerke verwendet, sondern Sie können die Elemente frei platzieren. Die Abarbeitungsliste umfasst folgende Elemente:

- Baustein
- Eingang
- Ausgang
- Sprung
- Label
- Return
- Kommentar

Die Ein- und Ausgänge dieser Elemente können Sie verbinden, indem Sie mit der Maus eine Verbindung zum nächsten Element ziehen. Die Verbindungslinie wird automatisch gezeichnet. Dabei wird unter Berücksichtigung der bestehenden Verbindungen die kürzeste Verbindungslinie gezeichnet. Wenn Sie Elemente verschieben, werden die Verbindungslinien automatisch angepasst. Kann eine Verbindungslinie aus Platzgründen nicht gezeichnet werden, so wird eine rote Linie zwischen Eingang und zugehörigem Ausgang dargestellt. Sobald genügend Platz vorhanden ist, wird diese Linie in eine Verbindungslinie umgewandelt.

Ein Vorteil des CFC-Editors im Vergleich zum Funktionsplan-Editor FUP ist, dass Rückkopplungen direkt eingefügt werden können.

Nähere Informationen über die Funktionsplan-Sprache finden Sie auf Seite 40 (→ Kap. 2.4.6, Freigrafischer Funktionsplaneditor CFC (CFC)).

Weitere Informationen finden Sie in den folgenden Abschnitten:

- (→ Abschnitt. Cursorpositionen im CFC), Seite 257
- (→ Abschnitt. [Einfügen] / [Baustein] im CFC), Seite 258
- (→ Abschnitt. [Einfügen] / [Eingang] im CFC), Seite 259
- (→ Abschnitt. [Einfügen] / [Ausgang] im CFC), Seite 259
- (→ Abschnitt. [Einfügen] / [Sprung] im CFC), Seite 259
- (→ Abschnitt. [Einfügen] / [Marke] im CFC), Seite 259
- (→ Abschnitt. [Einfügen] / [Return] im CFC), Seite 260
- (→ Abschnitt. [Einfügen] / [Kommentar] im CFC), Seite 260
- (→ Abschnitt. [Einfügen] / [Bausteineingang] im CFC), Seite 260
- (→ Abschnitt. [Einfügen] / [In-Pin], [Einfügen] / [Out-Pin] im CFC), Seite 260
- (→ Abschnitt. [Extras] / [Negieren] im CFC), Seite 260
- (→ Abschnitt. [Extras] / [Set/Reset] im CFC), Seite 261
- (→ Abschnitt. [Extras] / [EN/ENO] im CFC), Seite 261
- (→ Abschnitt. [Extras] / [Eigenschaften] im CFC), Seite 262
- (→ Abschnitt. Elemente markieren im CFC), Seite 263
- (→ Abschnitt. Elemente verschieben im CFC), Seite 263
- (→ Abschnitt. Elemente kopieren im CFC), Seite 263
- (→ Abschnitt. Verbindungen erstellen im CFC), Seite 263
- (→ Abschnitt. Verbindungen ändern im CFC), Seite 265
- (→ Abschnitt. Verbindungen löschen im CFC), Seite 265

- (→ Kap. [Extras] / [Verbindungsmarke] im CFC), Seite 265
- (→ Abschnitt. Inputs/Outputs "On the fly" einfügen), Seite 266
- (→ Abschnitt. Reihenfolge der Abarbeitung im CFC), Seite 266
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Anzeigen] im CFC), Seite 267
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Topologisch anordnen] im CFC), Seite 267
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Eins vor] im CFC), Seite 268
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Eins zurück] im CFC), Seite 269
- (→ Abschnitt. [Extras] / [Reihenfolge] / [An den Anfang] im CFC), Seite 269
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Ans Ende] im CFC), Seite 269
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Alles nach Datenfluss anordnen] im CFC), Seite 269
- (→ Abschnitt. [Extras] / [Makro erzeugen]), Seite 271
- (→ Abschnitt. [Extras] / [In Makro springen]), Seite 272
- (→ Abschnitt. [Extras] / [Makro expandieren]), Seite 272
- (→ Abschnitt. [Extras] / [Eine Makroebene zurück], [Extras] / [Alle Makroebenen zurück]), Seite 273
- (→ Abschnitt. Rückkopplungen), Seite 273
- (→ Abschnitt. [Extras] / [Zoom] im CFC), Seite 275
- (→ Abschnitt. [Extras] / [Instanz öffnen]), Seite 197
- (→ Abschnitt. CFC im Online-Modus), Seite 273

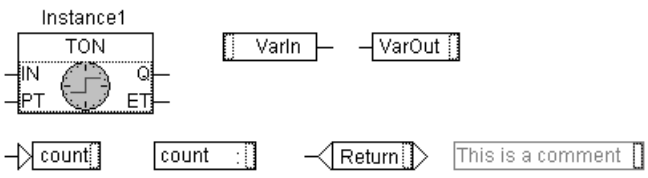
Cursorpositionen im CFC

Jede Textposition ist eine mögliche Cursorposition. Der markierte Text ist blau hinterlegt und kann geändert werden.

Die aktuelle Cursorposition ist durch ein gepunktetes Rechteck gekennzeichnet.

Mögliche Cursorpositionen im CFC:

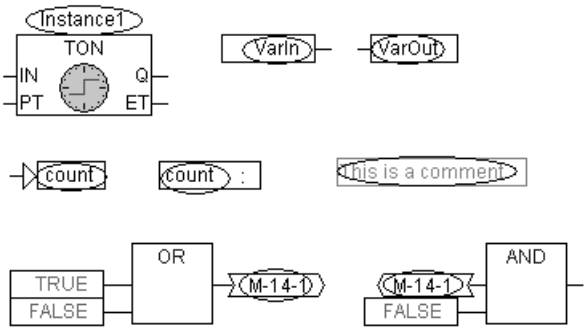
1. Rümpfe folgender Elemente:
   - Baustein
   - Eingang
   - Ausgang
   - Sprung
   - Label
   - Return
   - Kommentar



20413AXX

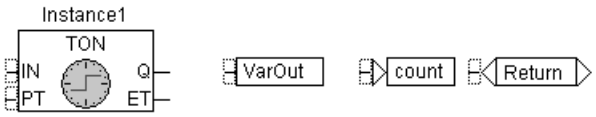
2. Textfelder folgender Elemente:
   - Baustein
   - Eingang
   - Ausgang
   - Sprung

- Label
- Return
- Verbindungsmarke



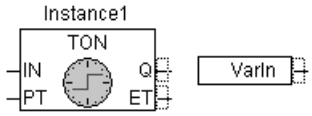
20414AXX

3. Eingänge folgender Elemente:
   - Baustein
   - Ausgang
   - Sprung
   - Return



20415AXX

4. Ausgänge folgender Elemente:
   - Baustein
   - Eingang



20416AXX

[Einfügen] / [Baustein] im CFC

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<B>

Mit dem Menübefehl [Einfügen] / [Baustein] ([Insert] / [Box]) fügen Sie Operatoren, Funktionen, Funktionsblöcke und Programme ein.

Zunächst fügen Sie mit diesem Menübefehl Sie einen AND-Operator ein. Klicken Sie auf den Operator und ersetzen Sie ihn durch den gewünschten Baustein, indem Sie den Text "AND" überschreiben.

Als Baustein können Sie verwenden:

- Operator
- Funktion
- Funktionsblock
- Programm

Mit der Eingabehilfe, die Sie mit der <F2>-Taste öffnen, können Sie den gewünschten Baustein aus der Liste der unterstützten Bausteine auswählen.

Wenn der neue Baustein eine andere Mindestanzahl von Eingängen hat, werden zu-

sätzliche Eingänge angehängt.

Wenn der neue Baustein eine kleinere Höchstzahl von Eingängen, werden die letzten Eingänge gelöscht.

[Einfügen] / [Eingang] im CFC

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<E>

Mit dem Menübefehl [Einfügen] / [Eingang] ([Insert] / [Input]) fügen Sie einen Eingang ein.

Der Eingang ist mit dem Text "???" vorbelegt. Klicken Sie auf diesen Text und ersetzen Sie ihn durch eine Variable oder eine Konstante. Dazu können Sie auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen.

[Einfügen] / [Ausgang] im CFC

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<A>

Mit dem Menübefehl [Einfügen] / [Ausgang] ([Insert] / [Output]) fügen Sie einen Ausgang ein.

Der Ausgang ist mit einem Standardvorbelegt. Klicken Sie auf diesen Text und ersetzen Sie ihn durch eine Variable ersetzen. Dazu können Sie auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen. Dabei wird dieser Variablen der Wert zugewiesen, der am Eingang des Ausgangs anliegt.

[Einfügen] / [Sprung] im CFC

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<J>

Mit dem Menübefehl [Einfügen] / [Sprung] ([Insert] / [Jump]) fügen Sie eine Sprung-Anweisung ein.

Die Sprung-Anweisung ist mit dem Text "???" vorbelegt. Klicken Sie auf diesen Text und ersetzen Sie ihn durch eine Sprungmarke, an die das Programm springen soll.

Eine Sprungmarke fügen Sie mit dem Menübefehl [Einfügen] / [Marke] ([Insert] / [Label]) ein.

[Einfügen] / [Marke] im CFC

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<L>

Mit dem Menübefehl [Einfügen] / [Marke] ([Insert] / [Label]) fügen Sie eine Sprungmarke ein.

Die Sprungmarke ist mit dem Text "???" vorbelegt. Klicken Sie auf diesen Text und ersetzen Sie ihn durch eine Sprungmarke. Im Online-Modus wird automatisch eine `Return`-Marke zur Markierung des Bausteinendes eingefügt.

Eine Sprung-Anweisung fügen Sie mit dem Menübefehl [Einfügen] / [Sprung] ([Insert] / [Jump]) ein.

[Einfügen] / [Return] im CFC

Schaltfläche: []

Tastaturbefehl: <Strg>+<R>

Mit dem Menübefehl [Einfügen] / [Return] ([Insert] / [Return]) fügen Sie eine Return-Anweisung ein.

Im Online-Modus wird automatisch eine Sprungmarke mit der Bezeichnung RETURN in der ersten Spalte nach dem letzten Element eingefügt. Beim Steppen springt das Programm vor dem Verlassen des Bausteins zu dieser Sprungmarke.

[Einfügen] / [Kommentar] im CFC

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<K>

Mit dem Menübefehl [Einfügen] / [Kommentar] ([Insert] / [Comment]) fügen Sie einen Kommentar ein.

Eine neue Zeile fügen Sie mit dem Tastaturbefehl <Strg>+<Eingabe> ein.

[Einfügen] / [Bausteineingang] im CFC

Tastaturbefehl: <Strg>+<U>

Mit dem Menübefehl [Einfügen] / [Bausteineingang] ([Insert] / [Input in Box]) fügen Sie eine Bausteineingang ein.

Die Zahl der Eingänge ist bei vielen Operatoren variabel (z.B. der Operator ADD kann zwei oder mehrere Eingänge haben).

Bevor Sie an einem Operator einen Eingang hinzufügen, müssen Sie der Operator (Cursorposition 1) markieren.

[Einfügen] / [In-Pin], [Einfügen] / [Out-Pin] im CFC

Schaltfächen: [ ] [ ]

Mit den Menübefehlen [Einfügen] / [In-Pin] bzw. [Einfügen] / [Out-Pin] ([Insert] / [In Pin]) bzw. ([Insert] / [Out Pin]) fügen Sie an einem Makro In- bzw. Out-Pins als Ein- und Ausgänge ein.

Diese Menübefehle stehen Ihnen zur Verfügung, wenn ein Makro zur Bearbeitung geöffnet ist.

In- bzw. Out-Pins unterscheiden sich von den normalen Ein- und Ausgängen der Bausteine durch die Darstellungsform und dadurch, das sie keinen Positionsindex erhalten.

[Extras] / [Negieren] im CFC

Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<N>

Mit dem Menübefehl [Extras] / [Negieren] ([Extras] / [Negation]) negieren Sie Eingänge, Ausgänge, Sprünge oder `Return`-Anweisungen. Das Symbol für die Negation ist ein kleiner Kreis auf einer Verbindungslinie.

- Wenn ein Eingang eines Bausteins, Ausgangs, Sprungs oder Returns markiert ist (Cursorposition 3), wird dieser Eingang negiert.
- Wenn ein Ausgang eines Bausteins oder Eingangs markiert ist (Cursorposition 4), wird dieser Ausgang negiert.

Eine Negation können Sie durch eine erneute Negation löschen.

[Extras] / [Set/Reset] im CFC

Schaltfläche: [ ]

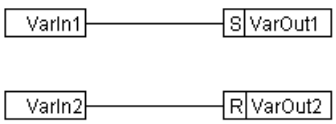
Tastaturbefehl: <Strg>+<T>

Mit dem Menübefehl [Extras] / [Set/Reset] ([Extras] / [Set/Reset]) wandeln Sie einen normalen Ausgang in einen Set-Ausgang und einen Set-Ausgang in einen Reset-Ausgang um.

Die markierte Position (Cursorposition 3) muss ein Eingang des Ausgangs sein.

Das Symbol für einen Set-Ausgang ist [S]. Das Symbol für einen Reset-Ausgang ist [R].

Beispiel



20441AXX

Die Variable `VarOut1` am Set-Ausgang wird auf TRUE gesetzt, wenn die Variable `VarIn1` auf TRUE gesetzt ist. Die Variable `VarOut1` bleibt auf TRUE gesetzt, selbst wenn die Variable `VarIn1` wieder auf FALSE zurückgesetzt wird.

Die Variable `VarOut2` am Reset-Ausgang wird auf FALSE gesetzt, wenn die Variable `VarIn2` auf TRUE gesetzt ist. Die Variable `VarOut2` bleibt auf FALSE gesetzt, selbst wenn die Variable `VarIn2` wieder auf FALSE zurückgesetzt wird.

Bei mehrfacher Ausführung des Menübefehls wandeln Sie einen markierten Ausgang zwischen Set-, Reset- und normalem Ausgang um.

[Extras] / [EN/ENO] im CFC

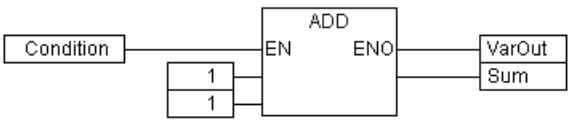
Schaltfläche: [ ]

Tastaturbefehl: <Strg>+<I>

Mit dem Menübefehl [Extras] / [EN/ENO] ([Extras] / [EN/ENO]) fügen Sie an einem markierten Baustein (Cursorposition 1) einen zusätzlichen booleschen Freigabe-Eingang `EN` (Enable In) und einen booleschen Ausgang `ENO` (Enable Out) ein.

Beispiele

Beispiel 1



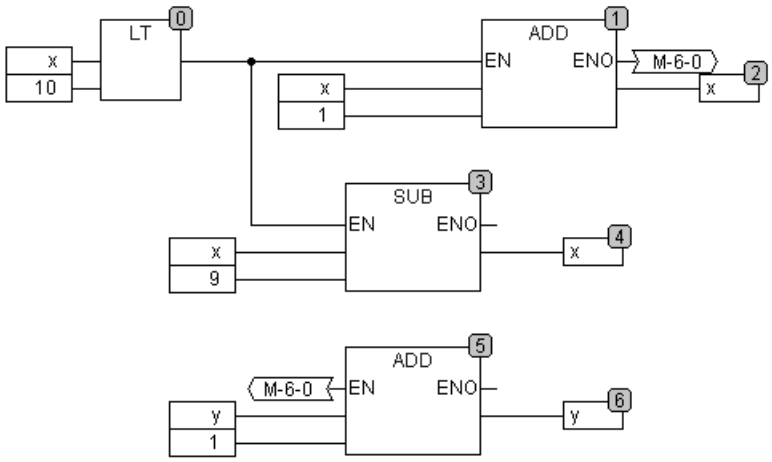
20417AXX

In diesem Beispiel wird der Baustein `ADD` nur ausgeführt, wenn die boolesche Variable `Condition` auf TRUE gesetzt ist. Der Ausgang `VarOut` wird nach der Ausführung des

Bausteins ADD ebenfalls auf TRUE gesetzt. Wenn die boolesche Variable Condition auf FALSE gesetzt wird, wird der Baustein ADD nicht mehr ausgeführt und der Ausgang VarOut behält den Wert TRUE!

Beispiel 2

Dieses Beispiel zeigt, wie Sie den Wert des Ausgangs ENO für weitere Bausteine verwenden können.



20418AXX

Der Eingang x ist mit 1 initialisiert, der Eingang y ist mit 0 initialisiert. Die Nummern in der rechten Ecke der Bausteine legen die Bearbeitungsreihenfolge fest.

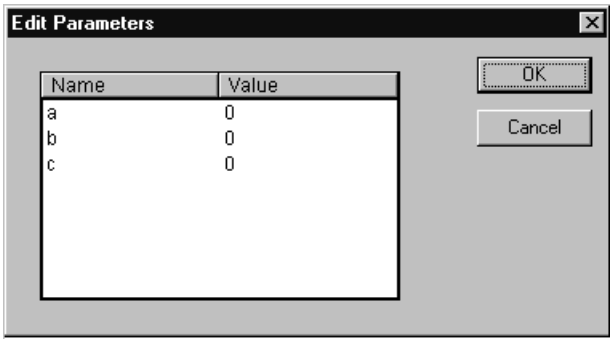
Der Eingang x wird solange um eins erhöht, bis x gleich 10 ist. Wenn x gleich 10 ist, werden die Bausteine SUB(3) und ADD(5) ausgeführt. Der Eingang x wird also auf den Wert 1 gesetzt und der Eingang y wird um 1 erhöht. Danach wird wieder der Baustein LT(0) ausgeführt, solange x kleiner als 10 ist.

Der Eingang y zählt also, wie oft x die Werte 1 bis 10 durchläuft.

[Extras] / [Eigenschaften] im CFC

Im CFC werden konstante Eingangsparameter, die als VAR_INPUT CONSTANT deklariert sind, von Funktionen und Funktionsblöcken nicht direkt dargestellt.

Wenn Sie den Rumpf eines Bausteins markieren (Cursorposition 1) und den Menübefehl [Extras] / [Eigenschaften] ([Extras] / [Properties]) ausführen oder auf den Rumpf doppelklicken, öffnet sich der Dialog [Parameter bearbeiten] ([Edit Parameters]). In diesem Dialog können Sie den Wert der Eingangsparameter ändern.



20442AEN

Wenn der Parameterwert in der Spalte [Wert] ([Value]) markiert ist, klicken Sie noch ein-

mal auf den Wert oder drücken Sie die <Leertaste>. Anschließend können Sie den Wert ändern. Bestätigen Sie die Änderung des Wertes mit der <Eingabe>-Taste. Wenn Sie die <Esc>-Taste drücken, werden die Änderungen nicht übernommen. Mit der <OK>-Taste werden alle Änderungen gespeichert.

Hinweis:

Diese Funktionalität und die Deklaration mit dem Schlüsselwort `VAR_INPUT CONSTANT` ist nur für den CFC-Editor von Bedeutung.

Im FUP-Editor werden immer alle Input-Variablen am Baustein angezeigt. Im FUP-Editor ist ohne Bedeutung, ob eine Variable mit dem Schlüsselwort `VAR_INPUT` oder `VAR_INPUT CONSTANT` deklariert ist.

Auch für die Texteditoren ist dies ohne Bedeutung.

Elemente markieren im CFC

Um ein Element zu markieren, klicken Sie mit der Maus auf den Rumpf des Elements (Cursorposition 1).

Um mehrere Elemente zu markieren, haben Sie folgende Möglichkeiten:

- Drücken Sie die <Umschalt>-Taste und klicken Sie mit der Maus nacheinander auf die entsprechenden Elemente.
- Ziehen Sie bei gedrückter linker Maustaste über den zu markierenden Elementen ein Fenster auf.

Mit dem Menübefehl [Extras] / [Alles Markieren] ([Extras] / [Select all]) markieren Sie alle Elemente.

Elemente verschieben im CFC

Um ein oder mehrere markierte Elemente zu verschieben, haben Sie folgende Möglichkeiten:

- Drücken Sie die <Umschalt>-Taste und verschieben Sie die Elemente mit den Pfeiltasten.
- Drücken Sie linke Maustaste und verschieben Sie die Elemente bei gedrückter Maustaste.

Wenn die Elemente andere Elemente nicht überdecken oder die vorgesehene Größe des Editors überschreiten, werden die Elemente beim Loslassen der linken Maustaste abgelegt. Wenn die markierten Elemente nicht abgelegt werden können, behalten sie ihre ursprünglichen Positionen und eine Warnung ertönt.

Elemente kopieren im CFC

Mit dem Menübefehl [Bearbeiten] / [Kopieren] ([Edit] / [Copy]) kopieren Sie ein oder mehrere markierte Elemente.

Mit dem Menübefehl [Bearbeiten] / [Einfügen] ([Edit] / [Paste]) fügen Sie die Elemente wieder ein.

Verbindungen erstellen im CFC

Den Eingang eines Elements können Sie mit genau einem Ausgang eines Elements

verbinden.

Den Ausgang eines Elements können Sie mit mehreren Eingängen von Elementen verbinden.

Beispiele

Um den Ausgang des Elementes E1 mit dem Eingang des Elements E2 zu verbinden, haben Sie folgende Methoden:

20443AXX

- Methode 1
  1. Klicken Sie mit der linken Maustaste auf den Ausgang des Elements E1 (Cursorposition 4) und halten Sie die Maustaste gedrückt.
  2. Ziehen Sie den Mauszeiger auf den Eingang des Elements E2 (Cursorposition 3).
  3. Lassen Sie die linke Maustaste dort los.
     Während des Ziehvorgangs wird eine Verbindung vom Ausgang des Elements E1 zum Mauszeiger gezeichnet.
- Methode 2
  1. Klicken Sie mit der linken Maustaste auf den Eingang des Elements E2 und halten Sie den Mauszeiger gedrückt.
  2. Ziehen Sie den Mauszeiger auf den Ausgang des Elements E1.
  3. Lassen Sie die linke Maustaste dort los.
- Methode 3
  1. Klicken Sie mit der linken Maustaste auf eines der Elemente E1 oder E2 (Cursorposition 1) und halten Sie den Mauszeiger gedrückt.
  2. Verschieben Sie das Element bis sich der Ausgang des Elements E2 und der Eingang des Elements E1 berühren.
  3. Lassen Sie die linke Maustaste dort los.
- Methode 4
  1. Wenn das Element E2 einen freien Eingang hat, ziehen Sie mit der Maus eine Verbindung vom Ausgang des Elements E1 in den Rumpf des Elements E2.
  2. Sobald Sie die Maustaste loslassen, wird automatisch eine Verbindung mit dem obersten freien Eingang des Elements E2 hergestellt.
     Wenn das Element E2 keinen freien Eingang hat, E2 jedoch ein Operator ist, der um einen Eingang erweitert werden kann, wird automatisch ein neuer Eingang erzeugt.

Mit Hilfe dieser Methoden können Sie auch den Ausgang eines Bausteins mit dem Eingang desselben Bausteins verbinden (Rückkopplung).

Um eine Verbindung zwischen zwei Pins herzustellen, gehen Sie folgendermaßen vor:

1. Klicken Sie mit der linken Maustaste auf einen Pin und halten Sie die linke Maustaste gedrückt.
2. Ziehen die Verbindung zum gewünschten Pin und lassen Sie Maustaste dort los.

Wenn Sie den Arbeitsbereich des Editors verlassen, während Sie die Verbindungs ziehen, wird der Bildbereich automatisch verschoben.

Während dem Sie die Verbindung herstellen, wird fr einfache Datentypen der Typ überprüft. Wenn die Typen der beiden Pins nicht kompatibel sind, nimmt der Cursor die Form eines durchgestrichenen Kreises an und die Verbindung wird nicht hergestellt. Für komplexe Datentypen erfolgt keine Überprüfung.

Verbindungen ändern im CFC

Beispiel

Die Verbindung zwischen dem Ausgang des Elements E1 und dem Eingang des Elementes E2 soll geändert werden in eine Verbindung zwischen dem Ausgang des Elements E1 und dem Eingang des Elements E3.

Gehen Sie dazu folgendermaßen vor:

1. Klicken Sie mit der Maus auf den Eingang des Elements E2 (Cursorposition 3) und halten Sie die Maustaste gedrückt.
2. Ziehen Sie den Mauszeiger auf den Eingang des Elements E3.
3. Lassen Sie die Maustaste dort los.

Verbindungen löschen im CFC

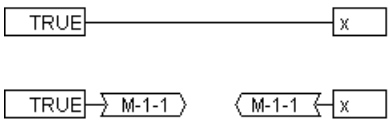
Um die Verbindung zwischen dem Ausgang des Elementes E1 und dem Eingang des Elementes E2 zu löschen, gibt es folgende Methoden:

- **Methode 1**
  1. Klicken Sie mit der Maus auf den Ausgang des Elements E1 (Cursorposition 4).
  2. Drücken Sie die <Entf>-Taste oder führen Sie auf den Menübefehl [Bearbeiten] / [Löschen] ([Edit] / [Delete]) aus.
     Wenn der Ausgang des Elements E1 mit mehreren Eingängen verbunden ist, werden alle Verbindungen gelöscht.
- **Methode 2**
  1. Klicken Sie mit der Maus auf den Eingang des Elements E2 (Cursorposition 4).
  2. Drücken Sie die <Entf>-Taste oder führen Sie auf den Menübefehl [Bearbeiten] / [Löschen] ([Edit] / [Delete]) aus.
- **Methode 3**
  1. Klicken Sie mit der Maus auf den Eingang des Elements E2 (Cursorposition 4) und halten Sie die Maustaste gedrückt.
  2. Ziehen Sie die Verbindung vom Eingang des Elements E2 weg.
  3. Wenn Sie die linke Maustaste in einem freien Bereich loslassen, wird die Verbindung gelöscht.

[Extras] / [Verbindungsmarke] im CFC

Verbindungen können statt mit Verbindungslinien auch mit Hilfe von Konnektoren (Verbindungsmarken) dargestellt werden. Dabei werden für den Ausgang und den zugehörigen Eingang je ein Konnektor eingefügt, der einen eindeutigen Namen hat.

Um eine Verbindung zwischen zwei Elementen in der Konnektor-Darstellung anzuzeigen, markieren Sie den Ausgang der Verbindungslinie (Cursorposition 3) und wählen den Menüpunkt [Extras] / [Verbindungsmarke] ([Extras] / [Connection mark]) aus.



20444AXX

Die Abbildung zeigt eine Verbindung vor und nach dem Menübefehl.

Standardmäßig vergibt das Programm einen eindeutigen Konnektornamen, der mit "M" beginnt. Der Konnektorname wird als Parameter des Ausgangs gespeichert, kann jedoch sowohl am Eingang als auch am Ausgang geändert werden:

Konnektornamen am Ausgang editieren M-1-1

Wenn Sie den Text im Konnektor ersetzen, wird der neue Konnektorname von allen zugehörigen Konnektoren an den Eingängen übernommen. Sie können jedoch keinen Name auswählen, der bereits an einer anderen Verbindungsmarke verwendet wird. Somit ist die Eindeutigkeit des Konnektornamens sichergestellt. Bei fehlerhafter Auswahl des Namens wird eine Meldung ausgegeben.

Konnektornamen am Eingang editieren M-1-1

Wenn Sie den Text im Konnektor ersetzen, wird der neue Konnektorname in der zugehörigen Verbindungsmarke am anderen Ausgang ersetzt. Verbindungen in der Konnektordarstellung können Sie wieder in gewöhnliche Verbindungen umgewandeln, indem Sie die Ausgänge der Verbindungen markieren (Cursorposition 4) und den Menüpunkt [Extras] / [Verbindungsmarke] ([Extras] / [Connection mark]) erneut auswählen.

Inputs/Outputs "On the fly" einfügen

Wenn nur ein Input- bzw. Output-Pin eines Elementes markiert ist, können Sie das entsprechende Input- bzw. Output-Element einfügen:

1. Geben Sie eine Zeichenfolge über die Tastatur eingeben
2. Drücken Sie die <Eingabe>-Taste
3. Geben Sie den Bezeichner im Editorfeld ein.

Reihenfolge der Abarbeitung im CFC

Beim CFC wird folgenden Elementen jeweils eine Abarbeitungsnummer zugeordnet:

- Ausgang
- Baustein
- Label
- Return
- Sprung

Im Online-Modus werden die einzelnen Elemente gemäß der Reihenfolge dieser Nummern abgearbeitet.

Beim Einfügen eines Elements wird die Nummer automatisch gemäß der topologischen Reihenfolge zugeordnet (von links nach rechts und von oben nach unten).

Wenn die Reihenfolge bereits geändert wurde, erhält das neue Element die Nummer seines topologischen Nachfolgers und alle höheren Nummern werden um eins erhöht.

Beim Verschieben eines Elements bleibt die Nummer erhalten.

Die Reihenfolge hat einen Einfluss auf das Ergebnis und muss in bestimmten Fällen geändert werden.

Wenn die Reihenfolge angezeigt wird, erscheint bei den Elementen in der rechten oberen Ecke die jeweilige Abarbeitungsnummer.

Weitere Informationen finden Sie in den folgenden Abschnitten:

- (→ Abschnitt. [Extras] / [Reihenfolge] / [Anzeigen] im CFC), Seite 267
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Topologisch anordnen] im CFC), Seite 267
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Eins vor] im CFC), Seite 268
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Eins zurück] im CFC), Seite 269

- (→ Abschnitt. [Extras] / [Reihenfolge] / [An den Anfang] im CFC), Seite 269
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Ans Ende] im CFC), Seite 269
- (→ Abschnitt. [Extras] / [Reihenfolge] / [Alles nach Datenfluss anordnen] im CFC), Seite 269

[Extras] / [Reihenfolge] / [Anzeigen] im CFC

Mit dem Menübefehl [Extras] / [Reihenfolge] / [Anzeigen] ([Extras] / [Order] / [Show Order]) schalten Sie im CFC die Anzeige der Abarbeitungsreihenfolge ein bzw. aus. Standardmäßig wird die Abarbeitungsreihenfolge angezeigt Der Menüpunkt ist dann mit einem Haken markiert.

Folgenden Elemente wird eine Abarbeitungsnummer zugeordnet, die jeweils in der rechten oberen Ecke des Elements angezeigt wird:

- Ausgang
- Baustein
- Label
- Return
- Sprung

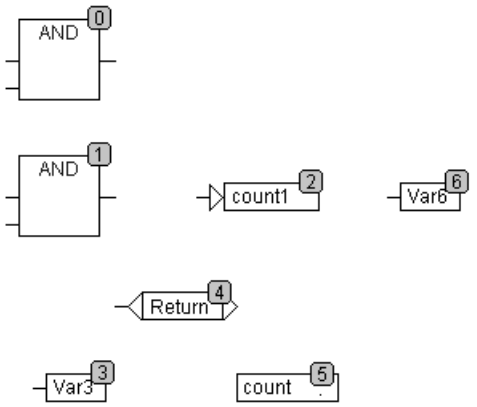
[Extras] / [Reihenfolge] / [Topologisch anordnen] im CFC

Im CFC sind die Elemente in topologischer Reihenfolge angeordnet, wenn die Abarbeitung der Elemente von links nach rechts und von oben nach unten stattfindet. Bei topologisch angeordneten Elementen erhöht sich die Abarbeitungsnummer von links nach rechts und von oben nach unten. Die Verbindungen spielen dann keine Rolle. Wichtig ist dabei nur die Position der Elemente.

Mit dem Menübefehl [Extras] / [Reihenfolge] / [Topologisch anordnen] ([Extras] / [Order] / [Order topologically]) ordnen Sie alle markierten Elemente topologisch an. Alle markierten Elemente werden dabei aus der Abarbeitungsliste herausgenommen. Danach werden die markierten Elemente einzeln von rechts unten nach links oben wieder in die verbleibende Abarbeitungsliste eingefügt.

Jedes markierte Element wird dabei in der Abarbeitungsliste vor dem topologischen Nachfolger eingefügt. Das heißt, es wird vor dem Element eingefügt, das bei einer topologischen Anordnung danach abgearbeitet würde, wenn alle Elemente des Editors in topologischer Reihenfolge angeordnet wären.

Beispiel



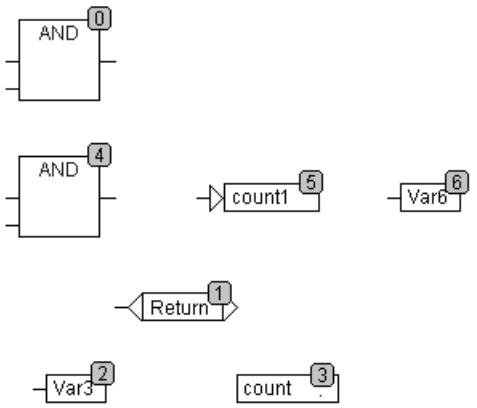
20419AXX

Die Elemente mit den Nummern 1 (AND), 2 (count1) und 3 (Var3) sind markiert.

Wenn Sie den Menübefehl [Topologisch anordnen] ([Order topologically]) ausführen, werden die Abarbeitungsnummern verändert:

- Die Abarbeitungsummer des Elements Var3 richtet sich nach der Abarbeitungsnummer des Elements count Da sich das Element Var3 links des Elements count befindet, liegt es in der Topologie vor dem Element count und erhält die Abarbeitungsnummer 2.
- Analog verhält es sich mit den Nummern für die Elemente AND und count1, die die Abarbeitungsnummern 4 und 5 erhalten. Deren neue Abarbeitungsnummern richten sich nach der Abarbeitungsnummer des Elements Var6.

Die Abarbeitungsreihenfolge sieht jetzt folgendermaßen aus:



20420AXX

Beim Ablegen eines neu erzeugten Bausteins wird dieser standardmäßig vor dessen topologischen Nachfolger in die Abarbeitungsliste eingeordnet.

[Extras] / [Reihenfolge] / [Eins vor] im CFC

Mit dem Menübefehl [Extras] / [Reihenfolge] / [Eins vor] ([Extras] / [Order] / [Order: one up]) verschieben Sie im CFC alle markierten Elemente innerhalb der Abarbeitungsreihenfolge um einen Platz nach vorne. Wenn eines der markierten Elemente das erste Element in der Abarbeitungsreihenfolge ist, wird dessen Abarbeitungsnummer nicht verändert.

[Extras] / [Reihenfolge] / [Eins zurück] im CFC

Mit dem Menübefehl [Extras] / [Reihenfolge] / [Eins zurück] ([Extras] / [Order] / [Order: one down]) verschieben Sie im CFC markierten Elemente innerhalb der Abarbeitungsreihenfolge um einen Platz nach hinten. Wenn eines der markierten Elemente das letzte Element in der Abarbeitungsreihenfolge ist, wird dessen Abarbeitungsnummer nicht verändert.

[Extras] / [Reihenfolge] / [An den Anfang] im CFC

Mit dem Menübefehl [Extras] / [Reihenfolge] / [An den Anfang] ([Extras] / [Order] / [Order:Start]) verschieben Sie im CFC alle markierten Elemente an den Anfang der Abarbeitungsreihenfolge. Dabei wird die Reihenfolge innerhalb der markierten Elemente nicht verändert. Die Reihenfolge innerhalb der nicht markierten Elemente wird ebenso nicht verändert.

[Extras] / [Reihenfolge] / [Ans Ende] im CFC

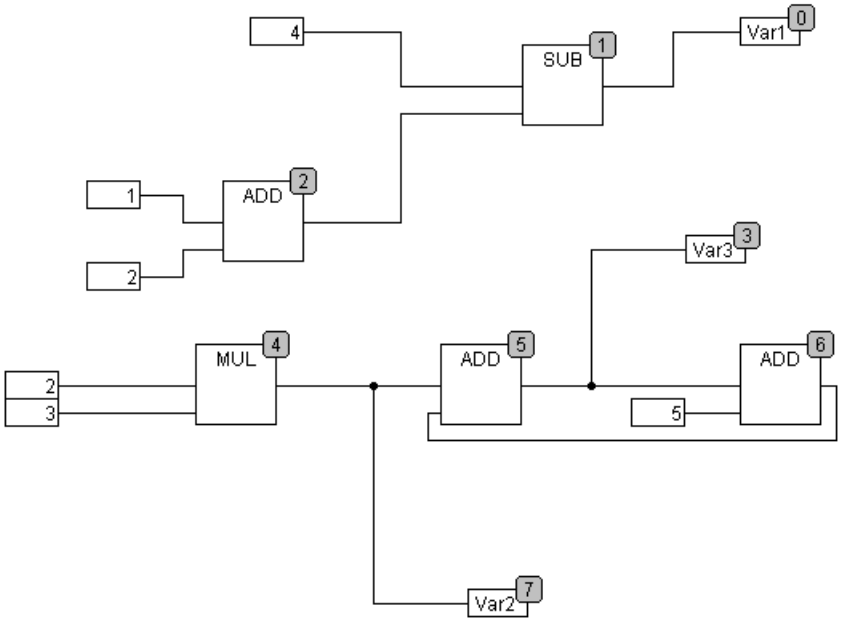
Mit dem Menübefehl [Extras] / [Reihenfolge] / [Ans Ende] ([Extras] / [Order] / [Order: End]) verschieben Sie im CFC alle markierten Elemente an das Ende der Abarbeitungsreihenfolge. Dabei wird die Reihenfolge innerhalb der markierten Elemente nicht verändert. Die Reihenfolge innerhalb der nicht markierten Elemente wird ebenso nicht verändert.

[Extras] / [Reihenfolge] / [Alles nach Datenfluss anordnen] im CFC

Der Menübefehl [Extras] / [Reihenfolge] / [Alles nach Datenfluß anordnen] ([Extras] / [Order] / [Order: everything according to the data flow]) wirkt auf alle Elemente des CFC. Nachdem Sie diesen Befehl ausgeführt haben, wird die Abarbeitungsreihenfolge vom Datenfluss der Elemente und nicht von deren Position festgelegt.

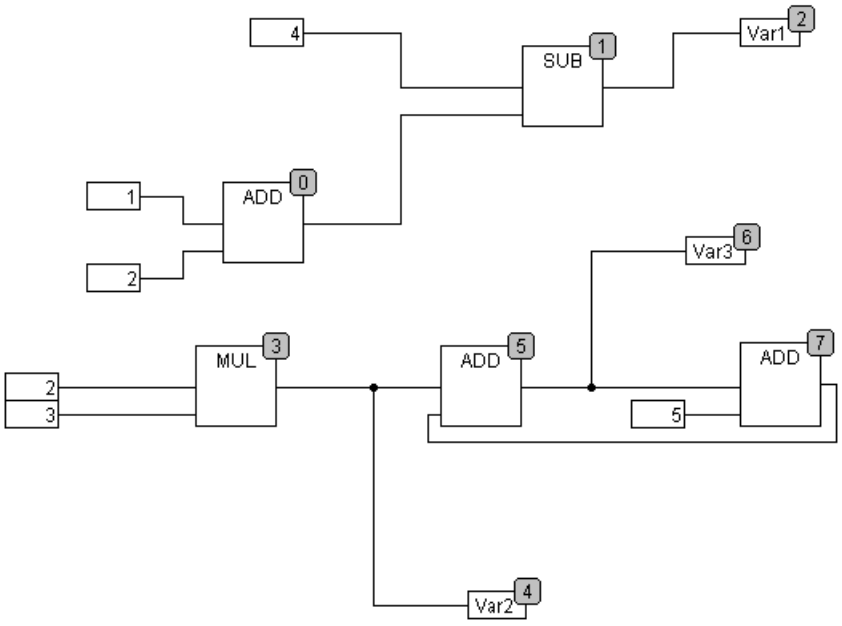
Beispiel

Die folgende Abbildung zeigt Elemente, die topologisch angeordnet sind.



20447AXX

Nachdem Sie den Menübefehl [Extras] / [Reihenfolge] / [Alles nach Datenfluß anordnen] ausgeführt haben, ergibt sich folgende Anordnung:



20448AXX

Wenn Sie den Menübefehl [Extras] / [Reihenfolge] / [Alles nach Datenfluß anordnen] ausführen, werden zunächst alle Elemente topologisch sortiert. Danach wird eine neue Abarbeitungsliste zusammengestellt. Ausgehend von den bekannten Werten der Eingänge wird ermittelt, welche der noch nicht nummerierten Elemente als nächstes abgearbeitet werden kann.

Im ersten Netzwerk kann z. B. der Baustein ADD sofort abgearbeitet werden, da die Werte an dessen Eingängen (1 und 2) bekannt sind. Erst danach kann der Baustein SUB abgearbeitet werden, da das Ergebnis von ADD bekannt sein muss usw.

Rückkopplungen werden als letztes eingefügt.

Die Reihenfolge gemäß dem Datenfluss hat folgenden Vorteil: Eine Ausgangsbox, die mit dem Out-Pin eines Bausteins verbunden ist, folgt immer unmittelbar auf diesen Baustein. Dies ist bei der topologischen Anordnung nicht immer der Fall.

Das Beispiel zeigt, dass die topologische Reihenfolge unter Umständen ein anderes Ergebnis als die Reihenfolge gemäß dem Datenfluss erzeugt.

[Extras] / [Makro erzeugen]

Schaltfläche: [ ]

Mit dem Menübefehl [Extras] / [Makro erzeugen] ([Extras] / [Create macro]) fassen Sie im freigrafischen Funktionsplan-Editor mehrere Bausteine, die gleichzeitig markiert sind, zu einem Block zusammen. Dieser Block ist ein Makro, dem Sie einen Namen geben können.

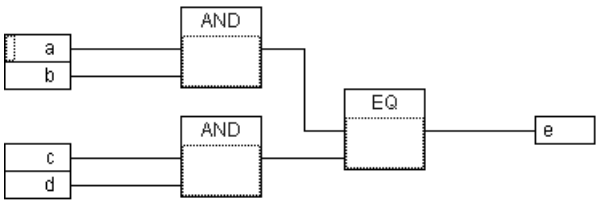
Makros können Sie nur mit Hilfe der Menübefehle [Kopieren] ([Copy]) und [Einfügen] ([Paste]) vervielfältigen. Dabei stellt jede Kopie ein eigenes Makro dar, dessen Namen Sie unabhängig auswählen können. Makros sind somit keine Referenzen.

Alle Verbindungen, die durch die Erzeugung des Makros getrennt werden, erzeugen In- bzw. Out-Pins am Makro. Verbindungen zu Inputs erzeugen einen In-Pin. Als Name neben dem In-Pin erscheint ein Default-Name der Form `In<n>`. Für Verbindungen zu Outputs erscheint `Out<n>`. Betroffene Verbindungen, die vor der Erzeugung des Makros Verbindungsmarken hatten, erhalten die Verbindungsmarke am Pin des Makros.

Ein Makro erhält zunächst den Namen "MAKRO". Diesen Namen können Sie anklicken und ändern. Der Name des Makros wird in der Titelleiste des Editorfensters hinter dem Bausteinnamen angezeigt.

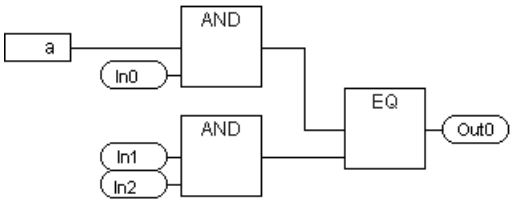
Beispiele

Selektion



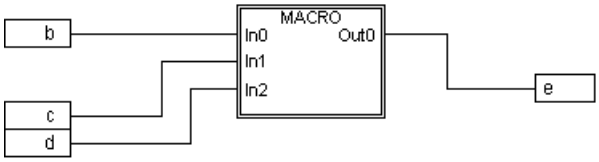
20449AXX

Makro



20500AXX

Im Editor



20501AXX

[Extras] / [In Makro springen]

Schaltfläche: [▷]

Mit dem Menübefehl [Extras] / [In Makro springen] ([Extras] / [Jump into macro]) oder mit einem Doppelklick auf den Rumpf des Makros öffnen Sie im freigrafischen Funktionsplan-Editor das Makro im Editorfenster des zugehörigen Bausteins. Hier können Sie das Makro bearbeiten. In der Titelleiste wird der Name des Makros hinter dem Namen des Bausteins angezeigt.

Die Pin-Boxen der Ein- und Ausgänge des Makros, die bei der Erstellung erzeugt wurden, können Sie wie die Ein- und Ausgänge von Bausteinen verschieben, löschen, hinzufügen. Sie unterscheiden sich nur in der Darstellung und haben keinen Positionsindex.

Mit den Schaltflächen [⇨] (Eingang) bzw. [⇦] (Ausgang) in der Symbolleiste fügen Sie die entsprechenden Pin-Boxen ein. Pin-Boxen haben abgerundete Ecken. Der Text der Pin-Box entspricht dem Namen des Pins in der Makrodarstellung.

Die Reihenfolge der Pins an der Makro-Box richtet sich nach der Abarbeitungsreihenfolge der Elemente des Makros: niedriger Reihenfolgeindex vor hohem, oberer Pin vor unterem.

Die Abarbeitungsreihenfolge innerhalb des Makros ist geschlossen. Das Makro wird als ein Block bearbeitet und zwar an der Position des Makros im übergeordneten Baustein. Die Befehle zur Manipulation der Reihenfolge wirken sich somit nur innerhalb des Makros aus.

[Extras] / [Makro expandieren]

Mit dem Menübefehl [Extras] / [Makro expandieren] ([Extras] / [Expand macro]) expandieren Sie im CFC das markierte Makro wieder. Die enthaltenen Elemente werden an der Position des Makros im Baustein eingefügt. Die Verbindungen zu den Pins des Makros werden wieder als Verbindungen zu den Ein- bzw. Ausgängen der Elemente dargestellt. Kann die Expansion des Makros aus Platzmangel nicht an der Position der Makrobox erfolgen, so wird das Makro solange nach rechts bzw. unten verschoben, bis genügend Platz zur Verfügung steht.

Hinweis:

Wenn Sie das Projekt unter der Projektversion 2.1 speichern, werden alle Makros ebenfalls expandiert.

Wenn Sie das Projekt in andere Sprachen konvertieren, werden alle Makros ebenfalls expandiert.

[Extras] / [Eine Makroebene zurück], [Extras] / [Alle Makroebenen zurück]

Schaltfächen: [ ] [ ]

Mit den Menübefehlen [Extras] / [Eine Makroebene zurück] ([Extras] / [Return to prior level]) bzw. [Eine Makroebene zurück] ([Return to top level]) können Sie zur darüberliegenden bzw. zur obersten Darstellungsebene von ineinandergeschachtelten Makros wechseln. Diese Schaltflächen stehen auch in der Symbolleiste zur Verfügung, wenn ein Makro zur Bearbeitung geöffnet ist.

Rückkopplungen

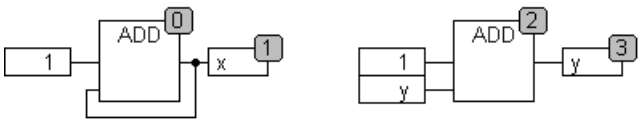
Der CFC kann im Gegensatz zum gewöhnlichen Funktionsplaneditor Rückkopplungen direkt darstellen. Für den Ausgang eines Bausteins wird generell eine interne Zwischenvariable angelegt.

Bei Operatoren ergibt sich der Datentyp der Zwischenvariable aus dem größten Datentyp der Eingänge.

Der Datentyp einer Konstanten ermittelt sich aus dem kleinstmöglichen Datentyp: Für die Konstante `1` wird der Datentyp `SINT` angenommen. Wird nun eine Addition mit Rückkopplung und der Konstanten `1` durchgeführt, so liefert der erste Eingang den Datentyp `SINT` und der zweite ist aufgrund der Rückkopplung undefiniert. Somit ist die Zwischenvariable auch vom Typ `SINT`. Der Wert der Zwischenvariable wird erst danach der Ausgangsvariablen zugewiesen.

Beispiel

Die folgende Abbildung zeigt eine Addition mit Rückkopplung und eine Aktion direkt mit einer Variablen. Die Variablen x und y sind vom Typ `INT`.



20502AXX

Zwischen den beiden Additionen bestehen folgende Unterschiede:

Die Variable y kann mit einem Wert ungleich 0 initialisiert werden, die Zwischenvariable der linken Addition jedoch nicht.

Die Zwischenvariable der linken Addition hat den Datentyp `SINT`, die der rechten den Datentyp `INT`. Die Variablen x und y haben ab dem 129. Aufruf unterschiedliche Werte. Die Variable x, obwohl vom Typ `INT`, erhält den Wert -127, weil die Zwischenvariable einen Überlauf hat. Die Variable y erhält dagegen den Wert 129.

CFC im Online-Modus

Überwachung (Monitoring)

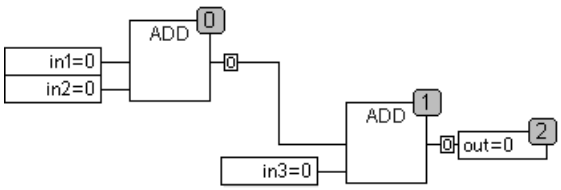
Die Werte der Eingänge und Ausgänge werden in den Input- bzw. Output-Boxen dargestellt. Konstanten werden nicht überwacht. Für nicht boolesche Variablen werden die Boxen entsprechend den angezeigten Werten vergrößert.

Der Variablenname und die Verbindunglinie von booleschen Verbindungen werden blau angezeigt, wenn der Wert auf TRUE gesetzt ist, ansonsten schwarz.

Interne boolesche Verbindungen werden im Online-Modus ebenfalls blau angezeigt, wenn der Wert auf TRUE gesetzt ist, ansonsten schwarz.

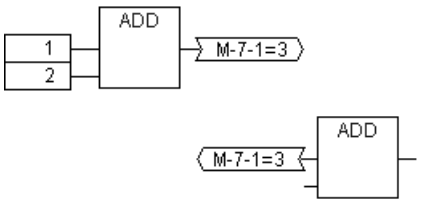
Der Wert von internen nicht booleschen Verbindungen wird in einer kleinen Box mit ab-

gerundeten Ecken angezeigt.



20503AXX

Pins in Makros werden wie Input- bzw. Output-Boxen überwacht.



20504AXX

Nicht boolesche Verbindungen mit Verbindungsmarken zeigen ihren Wert innerhalb der Verbindungsmarke an.

Für boolesche Verbindungen werden die Leitungen sowie die Markennamen ebenfalls blau dargestellt, wenn der Wert auf TRUE gesetzt ist, ansonsten schwarz.

Ablaufkontrolle

Bei aktivierter Ablaufkontrolle werden die durchlaufenen Verbindungen mit den in den Projektoptionen eingestellten Farben angezeigt.

Breakpoints

Einen Breakpoint können Sie auf alle Elemente setzen, die einen Index in der Abarbeitungsreihenfolge haben. Die Abarbeitung des Programms wird dann vor dem Ausführen des jeweiligen Elements gestoppt. Dies gilt für:

- Bausteine und Ausgänge vor dem Zuweisen der Eingänge
- für Sprungmarken vor dem Ausführen des Elements mit dem nächsten Index

Als Breakpoint-Position im Breakpoint-Dialog wird der Index des Elements verwendet.

Einen Breakpoint setzen Sie, indem Sie auf ein Element klicken und die <F9>-Taste drücken oder indem Sie den Menübefehl [Online] / [Breakpoint an/aus] ([Online] / [Step over]) ausführen. Wenn auf einem Element ein Breakpoint gesetzt wurde, wird dieser Breakpoint mit dem nächsten Ausführen des Befehls [Breakpoint an/aus] ([Toggle Breakpoint]) wieder gelöscht und umgekehrt. Zusätzlich kann ein Breakpoint durch Doppelklick auf das Element ein- und ausgeschaltet werden. Die Darstellung der Breakpoints erfolgt mit den in den Projekt-Optionen eingestellten Farben.

RETURN-Marke

Im Online-Modus wird in der ersten Spalte nach dem letzten Element im Editor automatisch eine Sprungmarke mit der Bezeichnung RETURN erzeugt. Diese Marke markiert das Ende des Bausteins. Beim Steppen springt das Programm vor dem Verlassen des Bausteins zur dieser Marke. In Makros werden keine RETURN-Marken eingefügt.

Steppen

Mit dem Menübefehl [Online] / [Einzelschritt über] ([Online] / [Step over]) springt das Programm jeweils zum Element mit dem nächsthöheren Index in der Abarbeitungsreihenfolge.

Wenn das aktuelle Element ein Makro oder ein Baustein ist, so verzweigt das Programm mit dem Menübefehl [Einzelschritt in] ([Step in]) in die Implementierung des Makros bzw. des Bausteins. Wenn Sie dort den Befehl [Einzelschritt über] ([Step over]) ausführen, springt das Programm auf das Element, das gemäß dem Index in der Abarbeitungsreihenfolge nach dem Makro steht.

[Extras] / [Zoom] im CFC

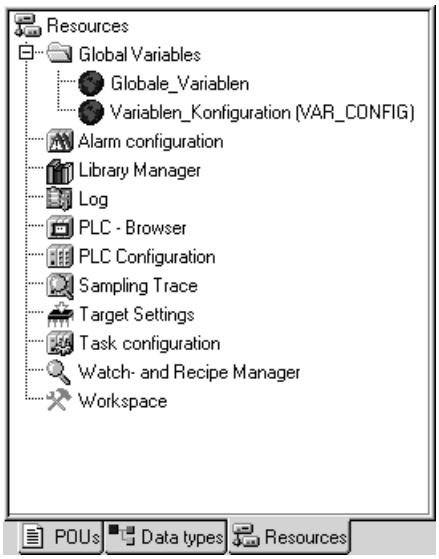
Tastaturbefehl: <Alt>+<Eingabe>

Mit dem Menübefehl [Extras] / [Zoom] ([Extras] / [Zoom]) öffnen Sie die Implementierung eines markierten Bausteins.

6 PLC-Editor Ressourcen

6.1 Übersicht Ressourcen

In der Registerkarte [Ressourcen] ([Resources]) des **Object Organizer** befinden sich die Objekte zum Organisieren der Projekte, zum Verfolgen von Variablenwerten und zum Konfigurieren der Steuerung.



20512AEN

Weitere Informationen finden Sie in den folgenden Kapiteln:

- (→ Kap. 6.2.3, Globale Variablen), Seite 277
- (→ Kap. 6.3, Arbeitsbereich), Seite 283
- (→ Kap. 6.4, Bibliotheksverwalter), Seite 284
- (→ Kap. 6.5, Steuerungskonfiguration), Seite 287
- (→ Kap. 6.6, Taskkonfiguration), Seite 304
- (→ Kap. 6.7, Traceaufzeichnung), Seite 310
- (→ Kap. 6.8, Watch- und Rezepturverwalter), Seite 320
- (→ Kap. 6.9, Zielsystem Einstellungen), Seite 325

Wenn ein Objekt der globalen Variablen geöffnet ist, können Sie eine Dokumentvorlage für ein Projekt erstellen und aufrufen. Mit Hilfe dieser Dokumentvorlage werden in der Dokumentation für dieselben Projektvariablen unterschiedliche Kommentare bereitgestellt.

6.2 Globale Variablen, Variablenkonfiguration, Dokumentvorlage

6.2.1 Objekte im Ordner [Globale Variablen]

In der Registerkarte [Ressourcen] ([Resources]) des **Object Organizer** befinden sich im Ordner [Globale Variablen] ([Global Variables]) standardmäßig die zwei folgenden Objekte. In Klammern stehen die vorbelegten Namen der Objekte.

- Globale Variablen ([Globale_Variablen])
- Variablenkonfiguration ([Variablen_Konfiguration])

Alle in diesen Objekten definierten Variablen sind im ganzen Projekt verfügbar.

Öffnen Sie den Ordner [Globale Variablen] und klicken Sie auf das entsprechende Objekt. Mit dem Menübefehl [Objekt bearbeiten] ([Edit Object]) öffnen Sie einen Dialog, in dem die bisher definierten globalen Variablenaufgelistet sind. Der Editor hierfür arbeitet wie der **Deklarationseditor**.

6.2.2 Mehrere Variablenlisten

Globale Projektvariablen `VAR_GLOBAL` und Variablenkonfigurationen `VAR_CONFIG` müssen in getrennten Objekten definiert werden.

Wenn Sie eine große Anzahl globaler Variablen deklariert haben, können Sie zur besseren Strukturierung neben der standardmäßig angelegten Liste [Globale_Variablen] weitere Variablenlisten wir folgt anlegen.

1. Klicken Sie in der Registerkarte [Ressourcen] ([Resources]) des Object Organizer auf den Ordner [Globale Variablen] [Global Variables] oder auf eines der bestehenden -Objekte mit globalen Variablen.
2. Um ein weiteres Objekt hinzufügen, führen Sie den Menübefehl [Projekt] / [Objekt] / [einfügen] ([Project] / [Object] / [Add]) aus.
3. Geben Sie dem Objekt im Dialog einen Namen. Bestätigen Sie mit [OK]. Ein weiteres Objekt mit dem Schlüsselwort `VAR_GLOBAL` wird angelegt. Mit [Abbrechen] ([Cancel]) brechen Sie den Vorgang ab.
4. Wenn Sie ein Objekt mit einer Variablenkonfiguration haben möchten, ändern Sie das Schlüsselwort in `VAR_CONFIG` um.

6.2.3 Globale Variablen

Was sind globale Variablen

Als globale Variablen können Sie Variablen, Konstanten oder remanente Variablen deklarieren, auf die im gesamten Projekt zugegriffen werden können soll.

Hinweis:

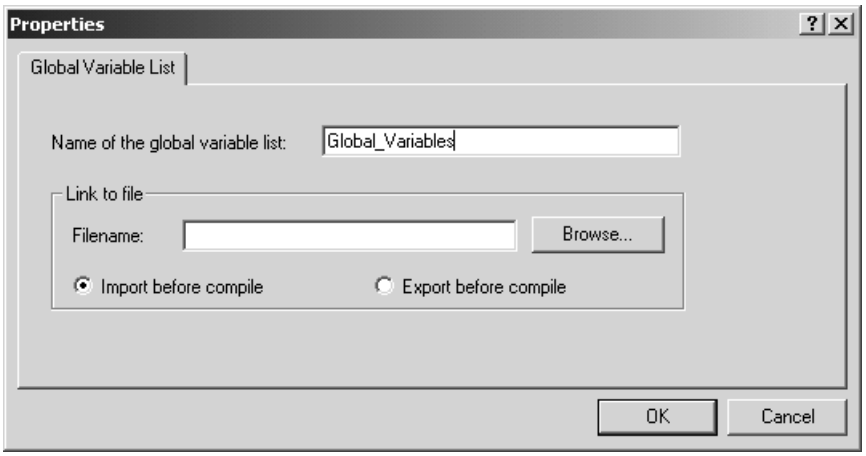
- Eine lokale Variable kann den gleichen Namen haben wie ein globale Variable. Innerhalb des Bausteins, in dem die lokale Variable definiert ist, stets die lokale Variable Vorrang.
- Zwei globale Variablen dürfen nicht den gleichen Namen haben.

  Wenn sowohl in der Steuerungskonfiguration als auch in einer globalen Variablenliste eine Variable mit dem gleichen Namen definiert ist, wird ein Übersetzungsfehler ausgegeben.

Anlegen einer Globalen Variablenliste

Um eine globale Variablenliste neu anzulegen, klicken Sie in der Registerkarte [Ressourcen] ([Resources]) des Object Organizer auf den Eintrag [Globale Variablen] ([Global Variables]) bzw. auf eine dort bereits angelegte globale Variablenliste.

Führen Sie anschließen den Menübefehl [Projekt] / [Objekt] / [Einfügen] ([Project] / [Object] / [Add]). Daraufhin wird der Dialog [Eigenschaften] ([Properties]) mit der Registerkarte [Globale Variablenliste] ([Global Variable List]) geöffnet.



20513AEN

Mit dem Menübefehl [Projekt] / [Objekt] / [Eigenschaften] ([Project] / [Object] / [Properties]) wird auch für die Konfiguration der im Object Organizer markierten globalen Variablenliste geöffnet.

[Name der globalen Variablenliste]	[Name of the global variable list]	Geben Sie einen neuen Listennamen ein.
[Dateiname]	[Filename]	Wenn bereits eine Exportdatei `*.exp` oder DCF-Datei zur Verfügung steht, die die gewünschten Variablen enthält, können Sie diese auswählen. Geben Sie dazu den entsprechenden Dateipfad ein oder öffnen Sie den Standarddialog [Textdatei auswählen] ([Select text file]) mit der Taste [Durchsuchen] ([Browse]) und wählen die Datei aus. DCF-Dateien werden beim Einlesen in die IEC-Syntax umgewandelt.
[Vor Übersetzen importieren]	[Import before compile]	Aktivieren Sie diese Option, wenn die angegebene externe Variablenliste vor dem Übersetzen des Projekts neu eingelesen werden soll.
[Vor Übersetzen exportieren]	[Export before compile]	Aktivieren Sie diese Option, wenn die Variablenliste vor dem Übersetzen des Projekts neu in die angegebene externe Datei geschrieben werden soll.

Schleißen Sie den Dialog [Eigenschaften] ([Properties]) [OK]. Das neue Objekt wird im Object Organizer angelegt und mit dem Symbol markiert.

Das neue Objekt können Sie anschließend öffnen, indem Sie den Menübefehl [Projekt] / [Objekt] / [Bearbeiten] ([Project] / [Object] / [Edit]) ausführen oder auf das Objekt doppelklicken.

Editieren der Listen für globale Variablen

Um den Editor für globale Variablen zu öffnen, doppelklicken Sie auf das Objekt [Globale Variablen] ([Global Variables]) im Object Organizer. Der Editor arbeitet wie der Deklarationseditor. Falls jedoch eine externe Variablenliste abgebildet wird, können Sie diese hier nicht mehr editieren. Externe Variablenlisten können nur extern bearbeitet werden und sie werden bei jedem Öffnen und bei jedem Übersetzen des Projekts neu

eingelesen.

Syntax

```
VAR_GLOBAL
   (* Variablendeklarationen *)
END_VAR
```

Editieren der Listen für remanente globale Variablen

Die Steuerung MOVI-PLC® bietet remanente Variablen an. Es gibt zwei Arten von remanenten globalen Variablen:

- **Retain-Variablen** behalten ihre Werte nach einem unkontrollierten Beenden des Laufzeitsystems (Aus/Ein) oder nach dem Menübefehl [Online] / [Reset].
- **Persistente Variablen** behalten ihre Werte nach einem Programm-Download.

Remanente Variablen erhalten zusätzlich das Schlüsselwort `RETAIN` und/oder `PERSISTENT`.

Achtung:

Persistente Variablen sind nicht immer auch Retain-Variablen!

Syntax

```
VAR_GLOBAL RETAIN
   (* Variablendeklarationen *)
END_VAR

VAR_GLOBAL PERSISTENT
   (* Variablendeklarationen *)
END_VAR
```

Globale Konstanten

Globale Konstanten verfügen zusätzlich über das Schlüsselwort `CONSTANT`.

Syntax

```
VAR_GLOBAL CONSTANT
   (* Variablendeklarationen *)
END_VAR
```

Hinweis:

Die Bezeichner von Konstanten dürfen nicht mit den nachfolgend aufgeführten Zeichenfolgen beginnen:

- MDX_
- MX_
- MC07_
- MM_

6.2.4 Variablenkonfiguration

Wie funktioniert die Variablenkonfiguration

In Funktionsbausteinen können bei Variablen, die zwischen den Schlüsselwörtern VAR und END_VAR definiert sind, Adressen für Eingänge und Ausgänge angegeben werden, die nicht vollständig definiert sind. Nicht vollständig definierte Adressen werden mit einem Stern (*) gekennzeichnet.

Beispiel

```
FUNCTION_BLOCK locio
   VAR
      loci AT %I*: BOOL := TRUE;
      loco AT %Q*: BOOL;
   END_VAR
```

In diesem Beispiel werden zwei lokale I/O-Variablen definiert, eine local-In-Variable %I* und eine local-Out-Variable %Q*.

Zur Konfiguration von lokalen Ein- und Ausgangs-Variablen steht Ihnen in der Registerkarte [Ressourcen] ([Resources]) des Object Organizer standardmäßig das Objekt -[Variablen_Konfiguration] (-[Variablen_Konfiguration]) zur Verfügung. Dieses Objekt können Sie umbenennen und Sie können weitere Objekte für die Variablenkonfiguration angelegen.

Der Editor für die Variablenkonfiguration arbeitet wie der Deklarationseditor.

Variablen zur lokalen Ein- und Ausgangs-Konfiguration müssen zwischen den Schlüsselwörtern VAR_CONFIG und END_VAR stehen.

Der Name einer lokalen Ein- oder Ausgangs-Variable besteht aus einem vollständigen Instanzpfad. Dabei sind die einzelnen Baustein- und Instanznamen durch Punkte voneinander getrennt. Die Deklaration muss eine Adresse enthalten, deren Klasse (Eingang/Ausgang) dem der nicht vollständig spezifizierten Adresse (%I*, %Q*) im Funktionsbaustein entspricht. Auch der Datentyp muss mit der Deklaration im Funktionsbaustein übereinstimmen.

Konfigurationsvariablen, deren Instanzpfad nicht gültig ist, weil die Instanz nicht existiert, werden als Fehler gekennzeichnet. Umgekehrt wird auch ein Fehler gemeldet, wenn für eine Instanzvariable keine Konfiguration existiert. Mit dem Menübefehl [Einfügen] / [Alle Instanzpfade] ([Insert] / [All instance paths]) öffnen Sie einen Dialog mit einer vollständigen Liste aller benötigten Konfigurationsvariablen.

Achtung:

Beim Einsatz der Steuerungskarte DHP11B werden bestimmte Speicherbereiche von %I, %Q und %M für die Steuerungskonfiguration belegt.

Beachten Sie dies bei der Konfiguration im Programm!

[Einfügen] / [Alle Instanzpfade]

Mit dem Menübefehl [Einfügen] / [Alle Instanzpfade] ([Insert] / [All instance paths]) wird ein Block VAR_CONFIG ... END_VAR erzeugt, der alle im Projekt vorhandenen Instanzpfade enthält. Um bestehende Adressen zu erhalten, werden bereits vorhandene Deklarationen nicht neu eingefügt.

Dieser Menübefehl steht im Dialog der Variablenkonfiguration zur Verfügung, wenn das Projekt mit dem Menübefehl [Project] / [Alles übersetzen] ([Project] / [Rebuild all]) übersetzt wurde.

6.2.5 Dokumentvorlage

Was ist eine Dokumentvorlage

Mit Ausnahme des Menübefehls [Projekt] / [In andere Sprache übersetzen] ([Project] / [Translate into other languages]) können Sie eine Dokumentvorlage verwenden, um ein Projekt mehrfach zu dokumentieren.

Verwenden Sie in folgenden Fällen eine Dokumentvorlage

- Sie benötigen für das Projekt eine Dokumentation mit verschiedensprachigen Kommentaren in den Variablen.
- Sie dokumentieren mehrere ähnliche Projekte, die dieselben Variablennamen benutzen.

Mit dem Menübefehl [Extras] / [Doku-Vorlage erstellen] ([Extras] / [Make Docuframe file]) erstellen Sie eine Dokumentvorlage. Dieser Menübefehl steht zur Verfügung, wenn eine globale Variablenliste geöffnet ist.

Die erstellte Datei können Sie mit einem beliebigen Texteditor öffnen und bearbeiten. Die Datei ist wie folgt aufgebaut:

- Die Datei beginnt mit der Zeile `DOKUFILE`.
- Anschließend folgt die Auflistung der Projektvariablen, wobei für jede Variable drei Zeilen vorgegeben sind.
- Anschließend folgt eine Zeile `VAR`, die anzeigt, dass eine neue Variable beginnt.
- Anschließend folgt eine Zeile mit dem Namen der Variablen.
- Am Ende folgt eine leere Zeile. Diese Zeile können Sie durch einen Kommentar zur Variablen ersetzen.

Variablen, die Sie nicht dokumentieren möchten, löschen Sie aus dem Text.

Sie können beliebig viele Dokumentvorlagen zu Ihrem Projekt erstellen.

20514AEN

Um eine Dokumentvorlage zu verwenden, führen Sie den Menübefehl [Extras] / [Doku-Vorlage auswählen] ([Extras] / [Link Docu file]) aus.

Wenn Sie das gesamte Projekt dokumentieren, oder Teile Ihres Projekts drucken, dann wird der Kommentar im Implementationsteil (nicht im Deklarationsteil!) mit ausgedruckt. Der Kommentartext, den Sie in der Dokumentvorlage für diese Variable erstellt haben, wird an der Stelle eingefügt, wo die Variable verwendet wird. Dieser Kommentar erscheint nur im Ausdruck!

[Extras] / [Doku-Vorlage erstellen]

Mit dem Menübefehl [Extras] / [Doku-Vorlage erstellen] ([Extras] / [Make Docuframe file]) erstellen Sie eine Dokumentvorlage. Dieser Befehl steht zur Verfügung, wenn eine globale Variablenliste geöffnet ist.

Nach dem Ausführen des Menübefehls öffnet sich der Dialog [Speichern unter] ([Save as]). Im Feld für den Dateinamen ist die Erweiterung `.txt` bereits eingegeben. Geben Sie einen beliebigen Namen ein. Bestätigen Sie mit [Speichern] ([Save]). Daraufhin wird eine Textdatei erstellt, in der sämtliche Variablen des Projekts aufgelistet sind.

[Extras] / [Doku-Vorlage auswählen]

Mit dem Menübefehl [Extras] / [Doku-Vorlage auswählen] ([Extras] / [Link Docu file]) wählen Sie eine Dokumentvorlage aus.

Es öffnet sich der Dialog zum Öffnen von Dateien. Wählen Sie die gewünschte Dokumentvorlage aus und klicken Sie auf die Taste [Öffnen] ([Open]). Wenn Sie das gesamte Projekt dokumentieren, oder Teile Ihres Projekts drucken, wird im Programmtext der Kommentar aller Variablen eingefügt. Dieser Kommentar erscheint nur im Ausdruck!

Zum Erstellen einer Dokumentvorlage verwenden Sie den Befehl [Extras] / [Doku-Vorlage erstellen] ([Extras] / [Make Docuframe file]).

6.3 Arbeitsbereich

Das Objekt [Arbeitsbereich] ([Workspace]) in der Registerkarte [Ressourcen] ([Resources]) enthält ein Abbild der eingestellten Projektoptionen. Wenn Sie das Objekt öffnen, erscheint der Dialog [Optionen] ([Options]) mit den bekannten Kategorien.

Weitere Informationen finden Sie auf Seite 73 (→ Kap. 4.2, Projektoptionen).

6.4 Bibliotheksverwaltung

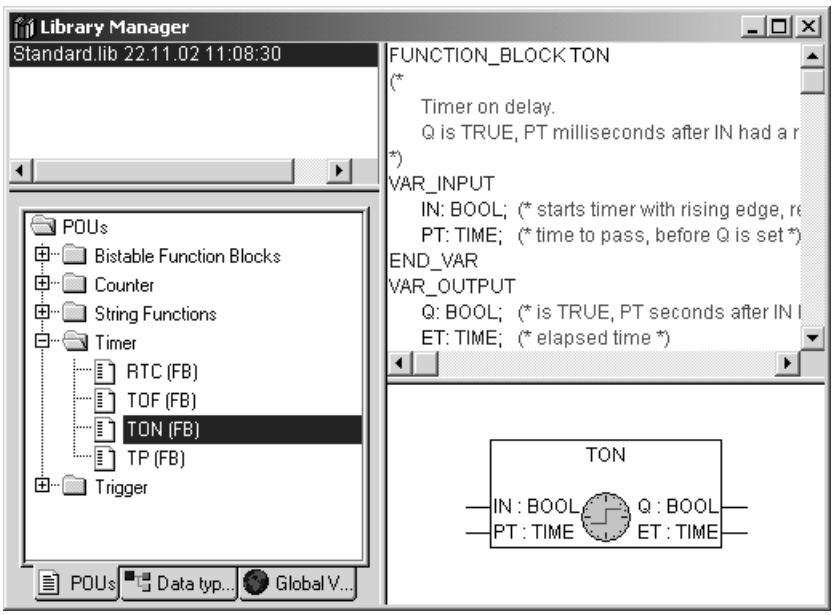
6.4.1 Bibliotheksverwalter

Der Bibliotheksverwalter zeigt alle Bibliotheken an, die an das aktuelle Projekt angeschlossen sind. Die Bausteine, Datentypen und globale Variablen der Bibliotheken können Sie wie selbstdefinierte Bausteine, Datentypen und globale Variablen verwenden.

Den Bibliotheksverwalter öffnen Sie mit dem Menübefehl [Fenster] / [Bibliotheksverwalter] ([Window] / [Library Manager]) oder mit einem Doppelklick auf [Bibliotheksverwalter] ([Library Manager]) in der Registerkarte [Ressourcen] ([Resources]) des **Object Organizer**. Die Information über die eingebundenen Bibliotheken wird mit dem Projekt abgespeichert: Wenn ein Objekt im Bibliotheksverwalter markiert ist, können Sie mit dem Menübefehl [Extras] / [Eigenschaften] ([Extras] / [Properties]) den Dialog mit der zugehörigen Information öffnen.

Zu den mit dem PLC-Editor erstellten Bibliotheken können Sie im Deklarationsteil Pragma-Anweisungen hinzufügen. Diese Pragma-Anweisungen bewirken, dass bei der Verwendung der Bibliothek in einem Projekt im Bibliotheksverwalter nicht der komplette Deklarationsteil dargestellt wird. Mit Pragma-Anweisungen können Sie einzelne Variablendeklarationen oder Kommentare vor dem Benutzer verbergen.

Beispiel



20515AEN

6.4.2 Bibliotheksverwalter nutzen

Das Fenster [Bibliotheksverwalter] ([Library Manager]) ist durch Bildschirmteiler in drei bzw. vier Bereiche aufgeteilt.

- Im linken oberen Bereich sind die Bibliotheken aufgelistet, die in das Projekt eingebunden sind.
- In dem darunter liegenden Bereich werden je nach gewählter Registerkarte die Bausteine, Datentypen oder globalen Variablen von der im oberen Bereich gewählten Bibliothek aufgelistet. Die Ordner in diesem Bereich können Sie mit einem Doppelklick auf die gewünschte Zeile auf- bzw. zuklappen. Vor zugeklappten Ordnern befindet sich ein Pluszeichen. Vor aufgeklappten Ordnern befindet sich ein Minuszeichen. Den gewünschten Baustein wählen Sie durch Mausklick oder mit den Pfeiltasten aus.
- Wenn ein Baustein ausgewählt ist, erscheint im rechten oberen Bereich des Bibliotheksverwalters die Deklaration des Bausteins.
- Wenn ein Baustein ausgewählt ist, erscheint im rechten unteren Bereich des Bibliotheksverwalters die grafische Darstellung des Bausteins als Blackbox mit Ein- und Ausgängen.

Wenn Sie Datentypen oder globale Variable anklicken ist der Bibliotheksverwalter ist durch Bildschirmteiler in drei Bereiche aufgeteilt. Der Bibliotheksverwalter zeigt im rechten Bereich die Deklaration an.

6.4.3 Standardbibliothek

Die Bibliothek `standard.lib` steht Ihnen standardmäßig zur Verfügung. Sie enthält alle Funktionen und Funktionsbausteine, die von der Norm IEC61131-3 für Standardbausteine eines IEC-Programmiersystems gefordert werden.

Der Unterschied zwischen einer Standardfunktion und einem Operator besteht darin, dass der Operator dem Programmiersystem implizit bekannt ist, während Sie die Standardbausteine in der Bibliothek `standard.lib` mit in das Projekt einbinden müssen.

Der Code zu diesen Bausteinen liegt als C-Bibliothek vor und ist Bestandteil des PLC-Editors.

6.4.4 Benutzerdefinierte Bibliotheken

Mit dem Menübefehl [Datei] / [Speichern unter] ([File] / [Save as]) können Sie ein Projekt als Bibliothek abspeichern. Das Projekt selbst bleibt unverändert, es wird zusätzlich eine Datei mit der Standarderweiterung `*.lib` erzeugt, die anschließend, wie z. B. die Standardbibliothek, unter dem eingegebenen Namen zur Verfügung steht.

Um die Bausteine eines Projektes in anderen Projekten verwenden zu können, wird es als interne Bibliothek `*.lib` gespeichert. Diese können Sie dann über den Bibliotheksverwalter in ein anderes Projekt einbinden.

Hinweis:

Beachten Sie, dass Sie mit Hilfe von Pragmas definieren können, welchern Teil der Deklaration der Bibliothek nach Einbinden der Bibliothek in ein Projekt im Bibliotheksverwalter angezeigt wird (Verbergen von Variablendeklarationen).

Das Einbinden von externen Bibliotheken wird nicht unterstützt.

6.4.5 [Einfügen] / [Weitere Bibliothek]

Mit dem Menübefehl [Einfügen] / [Weitere Bibliothek] ([Insert] / [Additional Library]) binden Sie eine weitere Bibliothek in Ihr Projekt ein.

Wenn Sie den Befehl ausführen, öffnet sich der Dialog zum Öffnen einer Datei. Hier können Sie die gewünschte Bibliothek mit der Erweiterung `*.lib` auswählen. Wenn Sie die Auswahl mit der Taste [Öffnen] ([Open]) bestätigen schließt sich der Dialog daraufhin und die Bibliothek wird im Bibliotheksverwalter aufgelistet.

Die Objekte der Bibliothek können Sie nun wie selbst definierte Objekte verwenden.

Bibliothekspfade

Beachten Sie, welche Bibliotheksverzeichnisse aktuell in den Projektoptionen definiert sind. Wenn Sie eine Bibliothek aus einem Verzeichnis einfügen, das dort nicht angegeben ist, wird die Bibliothek mit der entsprechenden Pfadangabe eingetragen.

Beispiel

Sie binden die Bibliothek `standard.lib` aus folgendem Verzeichnis ein:

[c:\programme\gemeinsame dateien\sew\targets\dhp11b\lib_movi-plc_basic_dhp11b\].

Wenn dieses Verzeichnis in den Projektoptionen definiert ist, befindet sich im Bibliotheksverwalter folgender Eintrag: [standard.lib <date and time of file>].

Beim Öffnen des Projekts werden die Bibliotheken, die im Bibliotheksverwalter eingetragenen sind, gesucht. Eine Bibliothek, die ohne Pfadangabe eingetragen ist, wird in den Bibliotheksverzeichnissen gesucht, deren Pfade in den Projektoptionen definiert sind.

6.4.6 Bibliothek entfernen

Mit dem Menübefehl [Bearbeiten] / [Löschen] ([Edit] / [Delete]) entfernen Sie die ausgewählte Bibliothek aus einem Projekt und dem Bibliotheksverwalter.

6.4.7 [Extras] / [Eigenschaften] im Bibliotheksverwalter

Mit dem Menübefehl [Extras] / [Eigenschaften] ([Extras] / [Properties]) öffnen Sie den Dialog [Informationen zu interner (bzw. externer) Bibliothek] ([Information about internal (or external) library]).

Für interne Bibliotheken enthält der Dialog die Daten, die beim Erstellen der Bibliothek als Projektinformationen eingegeben wurden, einschließlich der Statistik und der Lizenzinformationen.

Für externe Bibliotheken zeigt der Dialog den Namen und den Pfad der Bibliothek an.

6.5 Steuerungskonfiguration

6.5.1 Standardkonfiguration der Steuerung laden

Beim Anlegen eines neuen Projekts müssen Sie die Standardkonfiguration der Steuerung MOVI-PLC® laden.

Die Standardkonfiguration der Steuerung MOVI-PLC® laden Sie mit dem Menübefehl [Extras] / [Standardkonfiguration] ([Extras] / [Standard configuration]).

6.5.2 Allgemeine Einstellungen in der Steuerungskonfiguration

Wenn Sie im Konfigurationsbaum auf ein Organisationselement (z. B. [MOVI-PLC® Basic]) klicken, steht die Registerkarte [Einstellungen] ([Settings]) zur Verfügung:

[Adressen automatisch]	[Automatic calculation of addresses]	Jedes neu hinzugefügte Modul erhält automatisch eine Adresse, die sich aus der Adresse des zuvor eingefügten Moduls und dessen Größe ergibt. Wenn ein Modul aus der Konfiguration entfernt wird, werden die Adressen der nachfolgenden Module automatisch angepasst. Wenn Sie den Befehl [Extras] / [Adressen berechnen] ([Extras] / [Calculate addresses]) ausführen, werden die Adressen ab dem aktuell ausgewählten Modul neu ermittelt.
[Adressüberschneidungen prüfen]	[Check for overlapping addresses]	Adressüberschneidungen werden beim Übersetzen des Projekts überprüft und gemeldet.
[Konfigurationsdateien im Projekt speichern]	[Save configuration files in project]	Die Informationen der Konfigurationsdatei(en) `*.cfg` und der Gerätedateien, die der aktuellen Steuerungskonfiguration zugrunde liegen, werden im Projekt gespeichert. Somit bleibt die erstellte Konfiguration auch dann erhalten, wenn beim Öffnen des Projekts Konfigurationsdateien nicht gefunden werden. Dies gilt jedoch nicht, wenn durch die Konfigurationsdatei definiert ist, dass die Standardkonfiguration immer wieder hergestellt werden soll! Wenn diese Option nicht aktiviert ist, geht die gesamte projektspezifische Konfiguration verloren! Durch das Speichern der Konfigurationsinformationen im Projekt bleiben diese auch bei einem Zielsystem-Wechsel erhalten. Beachten Sie, dass zusätzlich die Konfigurationsdateien des Zielsystems berücksichtigt werden. Der globale Modus der Adressvergabe (flache Adressen / Adressen nach Id) innerhalb der Steuerungskonfiguration ist in der Konfigurationsdatei vordefiniert.

In der Registerkarte [Modulparameter] ([Module parameters]) stehen folgende Informationen zur Verfügung:

[PLC-Config Version]	[PLC-Config Version]	Der Wert zeigt die Version der auf dem PC installierten PLC-Konfiguration an, z. B. 18204848.51v1010.

6.5.3 Konfigurationsbaum

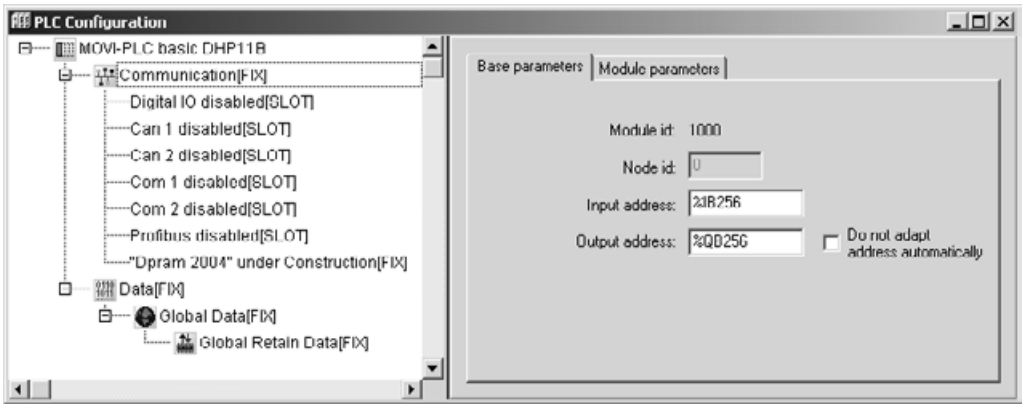
In der Steuerungskonfiguration wird z. B. das Organisationselement [MOVI-PLC basic DHP11B] als Konfigurationsbaum dargestellt.

Elemente, vor denen ein Pluszeichen steht, sind Organisationselemente, die Unterelemente enthalten.

Um das Organisationselement auzuklappen, doppelklicken Sie auf das Organisationslement.

Auf die gleiche Weise klappen Sie die Organisationselemente wieder zu.

Wenn Sie im Konfigurationsbaum ein Element anklicken, erscheint der Dialog [PLC-Konfiguration] ([PLC-Konfiguration]), in der Sie die Eigenschaften und Einstellungen des Elements ändern können.



20516AEN

Beim Modul [MOVI-PLC basic DHP11B] können Sie z. B. folgende Kanäle konfigurieren:

- Can 1
- Can 2
- Com 1
- Com 2
- Profibus
- Dpram

An die Kanäle können Sie folgende Unterelemente anhängen:

- MOVIDRIVE® MDXB
- MOVITRAC® B
- MOVIAXIS®
- CANopen IO Module
- SEW Processdata Module
- SCOM Receive
- SCOM Transmit

Die wichtigsten Befehle können Sie mit Hilfe des Kontextmenüs ausführen, das Sie mit einen Klick der rechten Maustaste öffnen.

Neuberechnen der Modul-Adressen, [Extras] / [Adressen berechnen]

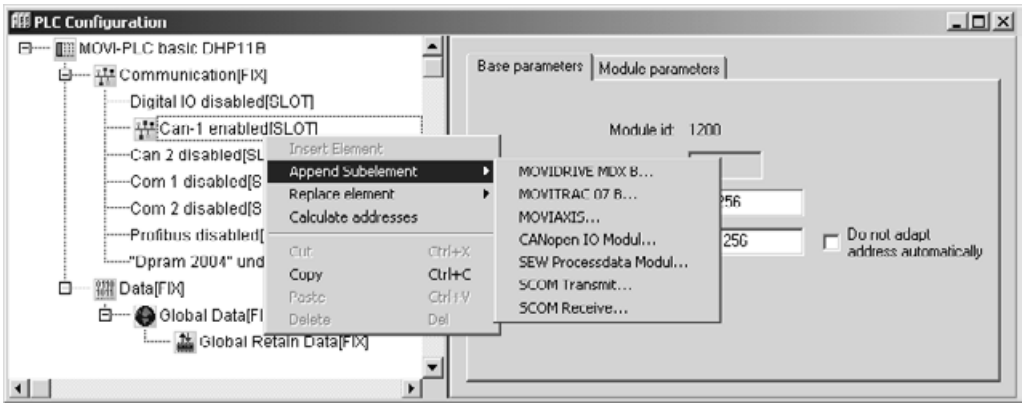
Mit dem Menübefehl [Extras] / [Adressen berechnen] ([Extras] / [Calculate addresses]) lassen Sie die Modul-Adressen neu berechnen. Die Adressen aller Elemente ab dem ausgewählten Modul werden nun neu berechnet. Dieser Menübefehl steht zur Verfügung, wenn die Option [Adressen automatisch] ([Automatic calculation of addresses]) in den Basisparametern der Steuerungskonfiguration aktiviert ist.

Zurück zur Standardkonfiguration, [Extras] / [Standardkonfiguration]

Mit dem Menübefehl [Extras] / [Standardkonfiguration] ([Extras] / [Standard configuration]) stellen Sie nach Änderungen im Konfigurationseditor die ursprüngliche Steuerungskonfiguration wieder her. Die ursprüngliche Steuerungskonfiguration basiert auf der Konfigurationsdatei `*.cfg` und ist im Projekt gespeichert.

Einfügen von Elementen, [Element einfügen], [Unterelement anhängen], [Element ersetzen]

In der Standardkonfiguration ist das Organisationselement [MOVI-PLC basic DHP11B] angelegt, an dem Sie weitere Elemente einfügen können.



20517AEN

Zum Einfügen weiterer Elemente stehen Ihnen im Kontextmenü, das Sie mit der rechten Maustaste öffnen, folgende Befehle zur Verfügung:

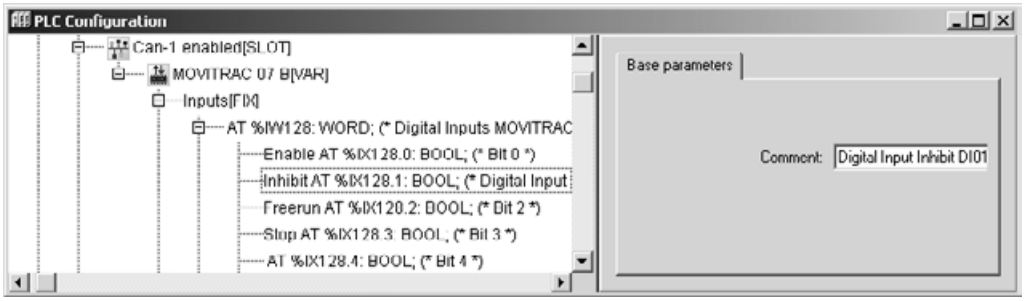
[Element einfügen]	[Insert Element]	Mit diesem Menübefehl fügen Sie ein neues Element vor dem ausgewählten Element ein. Das neue Element können Sie auswählen.
[Unterelement anhängen]	[Append Subelement]	Mit diesem Menübefehl hängen Sie ein neues Element als letztes Unterelement an das ausgewählte Element an. Das neue Element können Sie auswählen.
[Element ersetzen]	[Replace element]	Mit diesem Menübefehl ersetzen Sie das ausgewählte Element durch ein neues Element. Das neue Element können Sie auswählen. Beispiel: `Can 1 enabled` wird durch `Can 1 disabled` ersetzt.
[Adressen berechnen]	[Calculate Addresses]	Mit diesem Menübefehl lassen Sie die Adressen der Unterelemente des ausgewählten Elements neu berechnen. **Voraussetzung:** Die Option [Adressen nicht automatisch ändern] ([Do not adapt adresses automatically]) wurde nicht aktiviert.
[Ausschneiden]	[Cut]	Mit diesem Menübefehl löschen Sie das ausgewählte Element und speichern es in der Zwischenablage.
[Kopieren]	[Copy]	Mit diesem Menübefehl kopieren Sie das ausgewählte Element in der Zwischenablage.
[Einfügen]	[Paste]	Mit diesem Menübefehl fügen Sie das Element des Zwischenspeichers an der aktuellen Position ein.
[Löschen]	[Delete]	Mit diesem Menübefehl löschen Sie das ausgewählte Element.

Wenn es die Definition in der Konfigurationsdatei erlaubt, können Sie ein im Konfigurationsbaum ausgewähltes Element durch ein anderes Element ersetzen.

Mit dem Menübefehl [Extras] / [Element ersetzen] ([Extras] / [Replace element]) können Sie auch Kanäle umschalten, die als Ein- oder Ausgänge konfiguriert wurden.

Vergabe symbolischer Namen

Sie können jedem Element einen symbolischen Namen geben.


PLC Configuration
Can-1 enabled[SLOT]
MOVITRAC 07 B[VAR]
Inputs[FIX]
AT %IW128: WORD; (* Digital Inputs MOVITRAC
Enable AT %IX128.0: BOOL; (* Bit 0 *)
Inhibit AT %IX128.1: BOOL; (* Digital Input
Freerun AT %IX120.2: BOOL; (* Bit 2 *)
Stop AT %IX128.3: BOOL; (* Bit 3 *)
AT %IX128.4: BOOL; (* Bit 4 *)
Base parameters
Comment:
Digital Input Inhibit DI01

20518AEN

Um einem Element einen symbolischen Namen zu geben, gehen Sie wie folgt vor:

1. Klicken Sie mit der Maus bei dem gewünschten Element auf den Text "AT" der IEC-Adressenangabe. Es erscheint ein Eingabefeld.
2. Geben Sie den neuen symbolischen Namen ein und drücken Sie die <Eingabe>-Taste. Der eingegebene symbolische Name erscheint nun vor dem Text "AT".

Ebenso ändern Sie einen symbolischen Namen, indem Sie auf den symbolischen Namen doppelklicken und ihn im Eingabefeld ändern.

Hinweis:

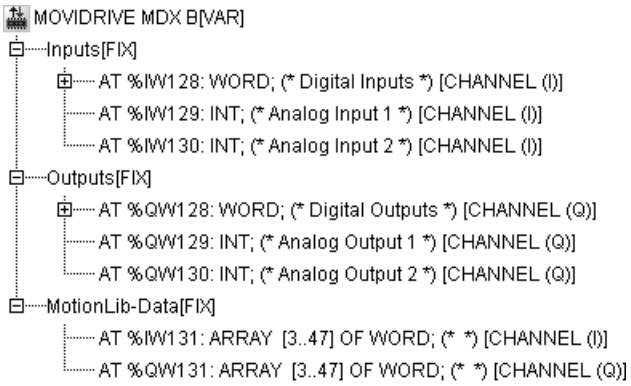
Die Vergabe eines symbolischen Namens entspricht der Variablendeklaration eines gültigen Projekts!

6.5.4 Übersicht der vorhandenen Unterelemente

Folgende Unterelemente können Sie in den Konfigurationsbaum einbauen:

- MOVIDRIVE® MDXB
- MOVITRAC® B
- MOVIAXIS®
- CANopen IO Modul
- SEW Processdata Modul
- SCOM Receive
- SCOM Transmit

MOVIDRIVE® MDXB


MOVIDRIVE MDX B[VAR]
Inputs[FIX]
AT %IW128: WORD; (* Digital Inputs *) [CHANNEL (I)]
AT %IW129: INT; (* Analog Input 1 *) [CHANNEL (I)]
AT %IW130: INT; (* Analog Input 2 *) [CHANNEL (I)]
Outputs[FIX]
AT %QW128: WORD; (* Digital Outputs *) [CHANNEL (Q)]
AT %QW129: INT; (* Analog Output 1 *) [CHANNEL (Q)]
AT %QW130: INT; (* Analog Output 2 *) [CHANNEL (Q)]
MotionLib-Data[FIX]
AT %IW131: ARRAY [3..47] OF WORD; (* *) [CHANNEL (I)]
AT %QW131: ARRAY [3..47] OF WORD; (* *) [CHANNEL (Q)]

20519AXX

Das Element MOVIDRIVE® MDXB können Sie wahlweise anhängen an:

- CAN 1-Strang
- CAN 2-Strang

Wenn Sie ein Unterelement anklicken, sind folgende Registerkarten verfügbar:

Registerkarte [Basisparameter] ([Base parameters])

[Modul-Id]	[Module id]	Die Modul-ID wird von der Software automatisch vergeben und angezeigt.
[Knotennummer]	[Node id]	Dieses Feld zeigt die fortlaufende Knotennummer des Unterelements an. Sie wird von der Software vergeben.
[Eingabeadresse]	[Input address]	Dieses Eingabefeld zeigt den %I-Bereich an, der automatisch berechnet wird.
[Ausgabeadresse]	[Output address]	Dieses Eingabefeld zeigt den %Q-Bereich an, der automatisch berechnet wird.
[Adressen nicht automatisch ändern]	[Do not adapt address automatically]	Wenn diese Option aktiv ist, verhindert die Software, dass die Adressen automatisch berechnet werden. Dies kann zu Fehlern bei Adressüberschneidungen führen.

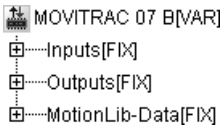
Hinweis:

Verwenden Sie geradzahlige Adressen, wenn Sie die Ein-/Ausgabeadressen manuell einstellen.

Registerkarte [Modulparameter] ([Module parameters])

[SBUS-Address]	[SBUS-Address]	Diese Zeile dient zur Einstellung der Adresse des Unterelements.
[Operation-Mode]	[Operation-Mode]	Diese Zeile dient zur Einstellung der Betriebsart des Unterelements. • Normal: Diese Betriebsart ist die Defaulteinstellung für die Kommunikation mit dem Umrichter der Motorachse. Bei Unterbrechung der Kommunikation zwischen der Steuerung MOVI-PLC® und einem Umrichter (z. B. MOVIDRIVE® MDX60B/61B) wechselt der Umrichter in den Zustand "Störung". • Debug: Diese Betriebsart stellt zusätzliche Debugging-Variablen zur Verfügung. Die Busüberwachung wird deaktiviert. • Simulation: Diese Betriebsart dient zur Simulation einer virtuellen Motorachse.
[I/O-Hold while Reset]	[I/O-Hold while Reset]	• TRUE: Bei Abbruch der Kommunikation bleibt der letzte gültige Wert der Eingänge erhalten. • FALSE: Bei Abbruch der Kommunikation werden die die Werte der Eingänge gelöscht.

MOVITRAC® B



20520AXX

Wenn Sie ein Unterelement anklicken, sind folgende Registerkarten verfügbar:

Registerkarte [Basisparameter] ([Base parameters])

[Modul-Id]	[Module id]	Die Modul-ID wird von der Software automatisch vergeben und angezeigt.
[Knotennummer]	[Node id]	Dieses Feld zeigt die fortlaufende Knotennummer des Unterelements an. Sie wird von der Software vergeben.
[Eingabeadresse]	[Input address]	Dieses Eingabefeld zeigt den %I-Bereich an, der automatisch berechnet wird.
[Ausgabeadresse]	[Output address]	Dieses Eingabefeld zeigt den %Q-Bereich an, der automatisch berechnet wird.
[Adressen nicht automatisch ändern]	[Do not adapt address automatically]	Wenn diese Option aktiv ist, verhindert die Software dass die Adressen automatisch berechnet werden. Dies kann zu Fehlern bei Adressüberschneidungen führen.

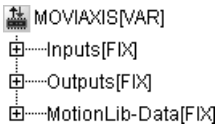
Hinweis:

Verwenden Sie geradzahlige Adressen, wenn Sie die Ein-/Ausgabeadressen manuell einstellen.

Registerkarte [Modulparameter] ([Module parameters])

[SBUS-Address]	[SBUS-Address]	Diese Zeile dient zur Einstellung der Adresse des Unterelementes.
[Operation-Mode]	[Operation-Mode]	Diese Zeile dient zur Einstellung der Betriebsart des Unterelements. • Normal: Diese Betriebsart ist die Defaulteinstellung für die Kommunikation mit dem Umrichter der Motorachse. Bei Unterbrechung der Kommunikation zwischen der Steuerung MOVI-PLC® und einem Umrichter (z. B. MOVIDRIVE® MDX60B/61B) wechselt der Umrichter in den Zustand "Störung". • Debug: Diese Betriebsart stellt zusätzliche Debugging-Variablen zur Verfügung. Die Busüberwachung wird deaktiviert. • Simulation: Diese Betriebsart dient zur Simulation einer virtuellen Motorachse.
[I/O-Hold while Reset]	[I/O-Hold while Reset]	• TRUE: Bei Abbruch der Kommunikation bleibt der letzte gültige Wert der Eingänge erhalten. • FALSE: Bei Abbruch der Kommunikation werden die die Werte der Eingänge gelöscht.

MOVIAXIS®



20521AXX

Wenn Sie ein Unterelement anklicken, sind folgende Registerkarten verfügbar:

Registerkarte [Basisparameter] ([Base parameters])

[Modul-Id]	[Module id]	Die Modul-ID wird von der Software automatisch vergeben und angezeigt.
[Knotennummer]	[Node id]	Dieses Feld zeigt die fortlaufende Knotennummer des Unterelements an. Sie wird von der Software vergeben.
[Eingabeadresse]	[Input address]	Dieses Eingabefeld zeigt den %I-Bereich an, der automatisch berechnet wird.
[Ausgabeadresse]	[Output address]	Dieses Eingabefeld zeigt den %Q-Bereich an, der automatisch berechnet wird.
[Adressen nicht automatisch ändern]	[Do not adapt address automatically]	Wenn diese Option aktiv ist, verhindert die Software, dass die Adressen automatisch berechnet werden. Dies kann zu Fehlern bei den Adressüberschneidungen führen.

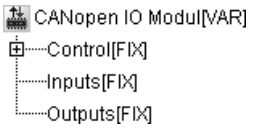
Hinweis:

Verwenden Sie geradzahlige Adressen, wenn Sie die Ein-/Ausgabeadressen manuell einstellen.

Registerkarte [Modulparameter] ([Module parameters])

[SBUS-Address]	[SBUS-Address]	Diese Zeile dient zur Einstellung der Adresse des Unterelementes.
[Operation-Mode]	[Operation-Mode]	Diese Zeile dient zur Einstellung der Betriebsart des Unterelements. • Normal: Diese Betriebsart ist die Defaulteinstellung für die Kommunikation mit dem Umrichter der Motorachse. Bei Unterbrechung der Kommunikation zwischen der Steuerung MOVI-PLC® und einem Umrichter (z. B. MOVIDRIVE® MDX60B/61B) wechselt der Umrichter in den Zustand "Störung". • Debug: Diese Betriebsart stellt zusätzliche Debugging-Variablen zur Verfügung. Die Busüberwachung wird deaktiviert. • Simulation: Diese Betriebsart dient zur Simulation einer virtuellen Motorachse.
[I/O-Hold while Reset]	[I/O-Hold while Reset]	• TRUE: Bei Abbruch der Kommunikation bleibt der letzte gültige Wert der Eingänge erhalten. • FALSE: Bei Abbruch der Kommunikation werden die die Werte der Eingänge gelöscht.

CANopen IO Modul



20522AXX

Wenn Sie ein Unterelement anklicken, sind folgende Registerkarten verfügbar:

Registerkarte [Basisparameter] ([Base parameters])

[Modul-Id]	[Module id]	Die Modul-ID wird von der Software automatisch vergeben und angezeigt.
[Knotennummer]	[Node id]	Dieses Feld zeigt die fortlaufende Knotennummer des Unterelements an. Sie wird von der Software vergeben.
[Eingabeadresse]	[Input address]	Dieses Eingabefeld zeigt den %I-Bereich an, der automatisch berechnet wird.
[Ausgabeadresse]	[Output address]	Dieses Eingabefeld zeigt den %Q-Bereich an, der automatisch berechnet wird.
[Adressen nicht automatisch ändern]	[Do not adapt address automatically]	Wenn diese Option aktiv ist, verhindert die Software dass die Adressen automatisch berechnet werden. Dies kann zu Fehlern bei den Adressüberschneidungen führen.

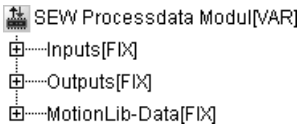
Hinweis:

Verwenden Sie geradzahlige Adressen, wenn Sie die Ein-/Ausgabeadressen manuell einstellen.

Registerkarte [Modulparameter] ([Module parameters])

[Adresse]	[Address]	Diese Zeile dient zur Einstellung der Adresse des Unterelementes.

SEW Processdata Modul



20523AXX

Wenn Sie ein Unterelement anklicken, sind folgende Registerkarten verfügbar:

Registerkarte [Basisparameter] ([Base parameters])

[Modul-Id]	[Module id]	Die Modul-ID wird von der Software automatisch vergeben und angezeigt.
[Knotennummer]	[Node id]	Dieses Feld zeigt die fortlaufende Knotennummer des Unterelements an. Sie wird von der Software vergeben.
[Eingabeadresse]	[Input address]	Dieses Eingabefeld zeigt den %I-Bereich an, der automatisch berechnet wird.
[Ausgabeadresse]	[Output address]	Dieses Eingabefeld zeigt den %Q-Bereich an, der automatisch berechnet wird.
[Adressen nicht automatisch ändern]	[Do not adapt the address automatically]	Wenn diese Option aktiv ist, verhindert die Software dass die Adressen automatisch berechnet werden. Dies kann zu Fehlern bei den Adressüberschneidungen führen.

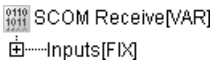
Hinweis:

Verwenden Sie geradzahlige Adressen, wenn Sie die Ein-/Ausgabeadressen manuell einstellen.

Registerkarte [Modulparameter] ([Module parameters])

[SBUS-Address]	[SBUS-Address]	Diese Zeile dient zur Einstellung der Adresse des Unterelementes.
[Processdata Count]	[Processdata Count]	Diese Zeile dient zur Einstellung der Prozessdatenwörter. Den Defaultwert von drei Prozessdatenwörter können Sie zwischen den Grenzenwerten Min/Max ändern.
[Operation-Mode]	[Operation-Mode]	Diese Zeile dient zur Einstellung der Betriebsart des Unterelements. • Normal: Diese Betriebsart ist die Defaulteinstellung für die Kommunikation mit dem Umrichter der Motorachse. Bei Unterbrechung der Kommunikation zwischen der Steuerung MOVI-PLC® und einem Umrichter (z. B. MOVIDRIVE® MDX60B/61B) wechselt der Umrichter in den Zustand "Störung". • Debug: Diese Betriebsart stellt zusätzliche Debugging-Variablen zur Verfügung. Die Busüberwachung wird deaktiviert.

SCOM Receive



20524AXX

Wenn Sie das Unterelement anklicken, sind folgende Registerkarten verfügbar:

Registerkarte [Basisparameter] ([Base parameters])

[Modul-ID]	[Module ID]	Die Modul-ID wird von der Software automatisch vergeben und angezeigt.
[Knotennummer]	[Node ID]	Dieses Feld zeigt die fortlaufende Knotennummer des Unterelements an. Sie wird von der Software vergeben.
[Eingabeadresse]	[Input address]	Dieses Eingabefeld zeigt den `%I`-Bereich an, der automatisch berechnet wird.
[Ausgabeadresse]	[Output address]	Dieses Eingabefeld zeigt den `%Q`-Bereich an, der automatisch berechnet wird.
[Adressen nicht automatisch ändern]	[Do not adapt address automatically]	Wenn diese Option aktiv ist, verhindert die Software, dass die Adressen automatisch berechnet werden. Dies kann zu Fehlern bei den Adressüberschneidungen führen.

Hinweis:

Verwenden Sie geradzahlige Adressen, wenn Sie die Ein-/Ausgabeadressen manuell einstellen.

Registerkarte [Modulparameter] ([Module parameters])

[SCOM-Id]	[SCOM-ID]	Diese Zeile dient zur Einstellung der Nutzdatenlänge beim Empfangen der SCOM-Objekte (Wertebereich: 0 ... 2047).
[Length]	[Length]	Diese Zeile dient zur Einstellung der Nutzdatenlänge beim Empfangen vom Sender der SCOM-Objekte. Den eingestellten Defaultwert von 8 Byte Nutzdaten können Sie zwischen den Grenzenwerten Min/ Max ändern.

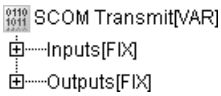
Hinweis zu SCOM-IDs:

Wenn die Bibliothek `MPLCMotion_xxx.lib` verwendet wird, sind bestimmte SCOM-IDs reserviert.

Hinweis:

Bei der Steuerungskarte MOVI-PLC® DHP11B muss beim Modulparameter `Length` der Wert "8 Byte" eingetragen werden, da immer alle Daten des Receive-Objekts empfangen werden.

SCOM Transmit



20525AXX

Wenn Sie das Unterelement anklicken, sind folgende Registerkarten verfügbar:

Registerkarte [Basisparameter] ([Basic parameters])

[Modul-Id]	[Module id]	Die Modul-ID wird von der Software automatisch vergeben und angezeigt.
[Knotennummer]	[Node id]	Dieses Feld zeigt die fortlaufende Knotennummer des Unterelements an. Sie wird von der Software vergeben.
[Eingabeadresse]	[Input address]	Dieses Eingabefeld zeigt den %I-Bereich an, der automatisch berechnet wird.
[Ausgabeadresse]	[Output address]	Dieses Eingabefeld zeigt den %Q-Bereich an, der automatisch berechnet wird.
[Adressen nicht automatisch ändern]	[Do not adapt address automatically]	Wenn diese Option aktiv ist, verhindert die Software dass die Adressenn automatisch berechnet werden. Dies kann zu Fehlern bei den Adressüberschneidungen führen.

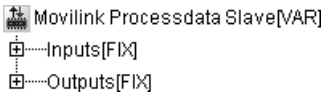
Hinweis:

Verwenden Sie geradzahlige Adressen, wenn Sie die Ein-/Ausgabeadressen manuell einstellen.

Registerkarte [Modulparameter] ([Module parameters])

[SCom-Id]	[SCom-ID]	Diese Zeile dient zur Einstellungder Nutzdatenlänge beim Empfangen der SCOM-Objekte (Wertebereich: 0 ... 2047).
[Length]	[Length]	Diese Zeile dient zur Einstellung der Nutzdatenlänge beim Empfangen vom Sender der SCOM-Objekte. Den eingestellten Defaultwert von 8 Byte Nutzdaten können Sie zwischen den Grenzenwerten Min/ Max ändern.
[CycleTime]	[CycleTime]	Diese Zeile dient zur Einstellung des Zeitintervalls, nachdem die Daten erneut gesendet werden. Den eingestellten Defaultwert 5 können Sie zwischen den Grenzenwerten Min/Max ändern.
[OffsetTime]	[OffsetTime]	Diese Zeile dient zur Verteilung der Buslast, wenn mehrere SCOM-Transmit-Objekte eingerichtet werden. Den eingestellten Defaultwert 0 können Sie zwischen den Grenzenwerten Min/Max ändern.

Movilink Processdata Slave



20526AXX

Das Element [Movilink Processdata Slave] können Sie wahlweise unter das Element [Movilink Master Protocol] folgender Stränge hängen:

- COM 1-Strang
- COM 2-Strang

Wenn Sie ein Unterelement anklicken, sind folgende Registerkarten verfügbar:

Registerkarte [Basisparameter] ([Basic parameters])

[Modul-Id]	[Module id]	Die Modul-ID wird von der Software automatisch vergeben und angezeigt.
[Knotennummer]	[Node id]	Dieses Feld zeigt die fortlaufende Knotennummer des Unterelements an. Sie wird von der Software vergeben.
[Eingabeadresse]	[Input address]	Dieses Eingabefeld zeigt den %I-Bereich an, der automatisch berechnet wird.
[Ausgabeadresse]	[Output address]	Dieses Eingabefeld zeigt den %Q-Bereich an, der automatisch berechnet wird.
[Adressen nicht automatisch ändern]	[Do not adapt address automatically]	Wenn diese Option aktiv ist, verhindert die Software dass die Adressen automatisch berechnet werden. Dies kann zu Fehlern bei den Adressüberschneidungen führen.

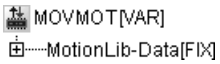
Hinweis:

Verwenden Sie geradzahlige Adressen, wenn Sie die Ein-/Ausgabeadressen manuell einstellen.

Registerkarte [Modulparameter] ([Module parameters])

[Address]	[Address]	Diese Zeile dient zur Einstellung der Adresse des Unterelements.
[PdCount]	[PdCount]	Diese Zeile dient zur Einstellung der Anzahl der Prozessdatenwörter (PD). Den Defaultwert von 3 PD können Sie zwischen den Grenzwerten 0 und 3 ändern.

MOVIMOT®



20527AXX

Das Unterelement [MOVIMOT®] können Sie wahlweise unter das Element [Movilink Master Protocol] folgender Stränge hängen:

- COM 1-Strang
- COM 2-Strang

Wenn Sie ein Unterelement anklicken, sind folgende Registerkarten verfügbar:

Registerkarte [Basisparameter] ([Base parameters])

[Modul-Id]	[Module id]	Die Modul-ID wird von der Software automatisch vergeben und angezeigt.
[Knotennummer]	[Node id]	Dieses Feld zeigt die fortlaufende Knotennummer des Unterelements an. Sie wird von der Software vergeben.
[Eingabeadresse]	[Input address]	Dieses Eingabefeld zeigt den %I-Bereich an, der automatisch berechnet wird.
[Ausgabeadresse]	[Output address]	Dieses Eingabefeld zeigt den %Q-Bereich an, der automatisch berechnet wird.
[Adressen nicht automatisch ändern]	[Do not adapt address automatically]	Wenn diese Option aktiv ist, verhindert die Software dass die Adresse automatisch berechnet wird. Dies kann zu Fehlern bei den Adressüberschneidungen führen.

Hinweis:

Verwenden Sie geradzahlige Adressen, wenn Sie die Ein-/Ausgabeadressen manuell einstellen.

Registerkarte [Modulparameter] ([Module parameters])

[Address]	[Address]	Diese Zeile dient zur Einstellung der Adresse des Unterelements.
[Operation-Mode]	[Operation-Mode]	Diese Zeile dient zur Einstellung der Betriebsart des Unterelements. • Normal: Diese Betriebsart ist die Defaulteinstellung für Kommunikation mit dem Umrichter der Motorachse. Bei Unterbrechung der Kommunikation zwischen der Steuerung MOVI-PLC® und einem Umrichter (z. B. MOVIMOT®) wechselt der Umrichter in den Zustand "Störung". • Debug: Diese Betriebsart stellt zusätzliche Debugging-Variablen zur Verfügung. Die Busüberwachung wird deaktiviert.

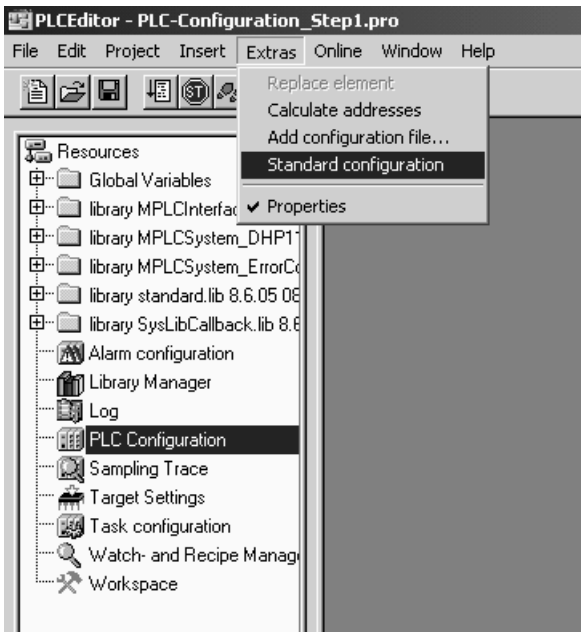
6.5.5 Beispiel einer Steuerungskonfiguration

Im folgenden Beispiel soll ein Antriebsumrichter MOVIDRIVE® MDX61B an den CAN 1-Strang angeschlossen werden.

Gehen Sie dazu bitte wie folgt vor:

Standardkonfiguration laden

1. Öffnen Sie ein neues Projekt.
2. Klicken Sie im **Object Organizer** auf die Registerkarte [Ressourcen] ([Resources]).
3. Doppelklicken Sie in der Registerkarte [Ressourcen] ([Resources]) auf das Objekt [Steuerungskonfiguration] ([PLC Configuration]).
4. Laden Sie die Standardkonfiguration mit dem Menübefehl [Extras] / [Standardkonfiguration] ([Extras] / [Standard configuration]).



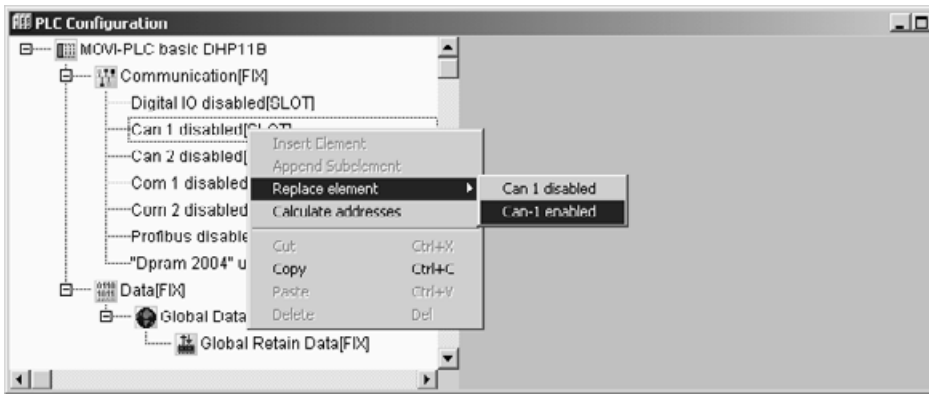
20528AEN

CAN 1 Element aktivieren und Unterelement MOVIDRIVE® B anhängen

In der Steuerungskonfiguration können Sie verschiedene Elemente an die zur Verfügung stehenden Schnittstellen hängen.

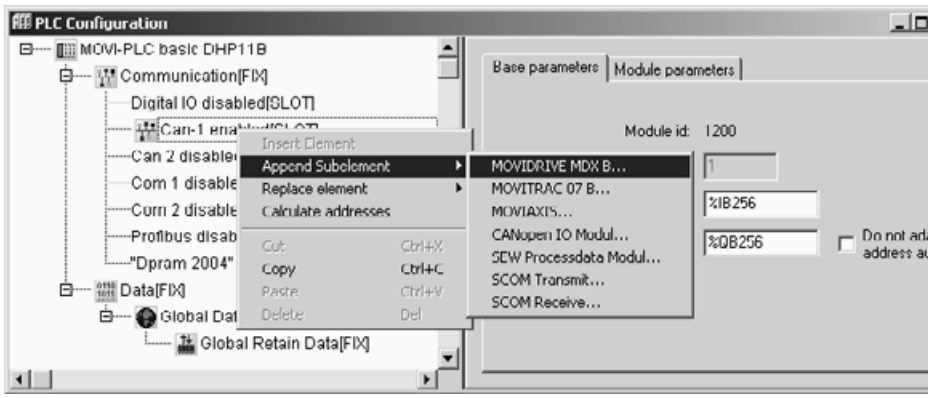
Im folgenden Beispiel wird das Element MOVIDRIVE® MDX61B an den Can 1-Strang angehängt.

1. Doppelklicken Sie auf das Element [Steuerungskonfiguration] ([PLC Configuration]). Auf der rechten Seite erscheint der Dialog [Steuerungskonfiguration] ([PLC Configuration]).
2. Klappen Sie in der Steuerungkonfiguration die Unterelemente auf, indem Sie auf die Pluszeichen klicken.
3. Klicken Sie mit der rechten Maustaste auf das Element [Can 1 disabled] und wählen Sie im Kontextmenü [Element ersetzen] / [Can-1 enabled] ([Replace element] / [Can-1 enabled]) aus. Der Can 1 Strang wird freigegeben.



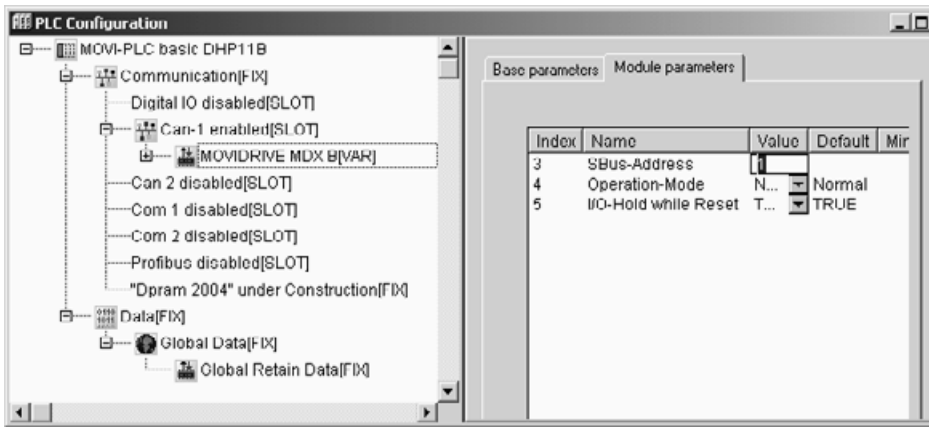
20529AEN

4. Klicken Sie mit der rechten Maustaste auf das Element [Can-1 enabled] ([Can-1 enabled]) und wählen Sie im Kontextmenü [Unterelement anhängen] / [MOVIDRIVE MDX B] ([Append Subelement] / [MOVIDRIVE MDX B] aus. Das Unterelement MOVIDRIVE MDX B wird angehängt.



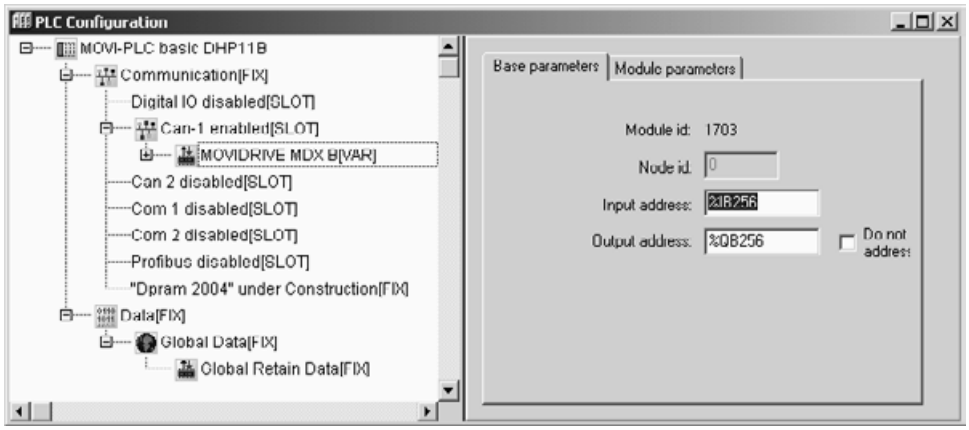
20530AEN

5. Klicken Sie auf die Registerkarte [Modulparameter] ([Module parameters]). Geben Sie in der Zeile [SBus-Address] die physikalische Adresse des Antriebsumrichters MOVIDRIVE® MDX61B ein.



20531AEN

6. In der Registerkarte [Basisparameter] ([Base parameters]) können Sie die automatisch eingestellten Basisparameter des Antriebsumrichters MOVIDRIVE® MDX61B z. B. für die Ein- und Ausgabeadressen betrachten.

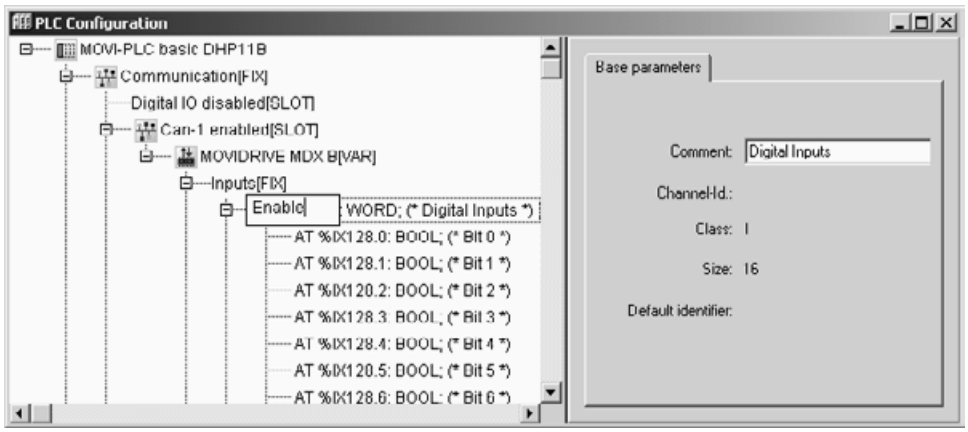


20532AEN

7. Bei Bedarf geben Sie den Unterelementen [Inputs] symbolische Namen.

   Klappen Sie dazu alle Unterelemente auf, indem Sie jeweils auf die Pluszeichen klicken.

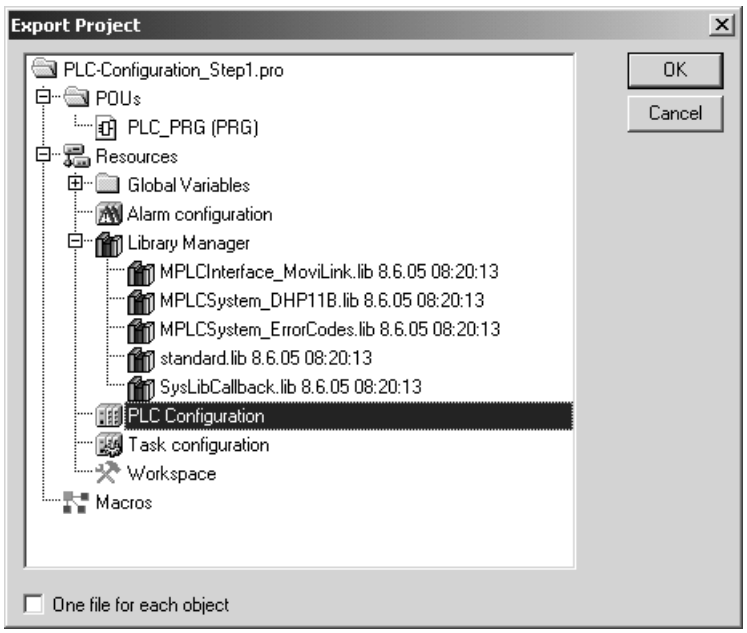
   Klicken Sie mit der rechten Maustaste auf den Text "AT" des gewünschten Unterelements, z. B. Eingangsbit 0 der Eingangsklemmen und ändern Sie den Namen und den Kommentar des Unterelements.



20533AEN

8. Exportieren Sie die festgelegte Steuerungskonfiguration, z. B. um sie an weitere Projekte übertragen zu können.

   Führen Sie dazu den Menübefehl [Projekt] / [Exportieren] ([Project] / [Export]) aus. Der Dialog [Projekt exportieren] ([Export Project]) wird geöffnet.



20534AEN

   Klicken in diesem Dialog auf das Objekt [Steuerungskonfiguration] ([PLC Configuration]) und bestätigen Sie mit [OK].

   Geben Sie im neuen Dialog Namen und Pfad der zu exportierenden Datei ein.

   Die exportierte Datei steht damit z. B. für den Import in weitere Projekte zur Verfügung.

6.6 Taskkonfiguration

6.6.1 Überblick

Im Allgemeinen wird die Abarbeitung eines Projekts mit Hilfe des speziellen Programms PLC_PRG gesteuert. Zusätzlich kann die Abarbeitung eines Projekts von der Taskverwaltung gesteuert werden.

Eine Task ist die zeitliche Ablaufeinheit eines IEC-Programms. Sie ist definiert durch einen Namen, eine Priorität und einen Typ, der festlegt, welche Bedingung den Start der Task auslöst. Diese Bedingung kann entweder zeitlich definiert sein (Zyklusintervall, freilaufend) oder durch ein internes oder ein externes Ereignis, bei dessen Eintreten die Task ausgeführt werden soll. Beispielsweise kann die Task bei einer steigenden Flanke, einer globalen Projektvariablen oder einem Interrupt-Event der Steuerung ausgeführt werden.

Jede Task kann eine Folge von Programmen zugeordnet werden, die beim Ausführen die Task abgearbeitet werden.

Durch Zusammenwirken von Priorität und Bedingung wird festgelegt, in welcher zeitlichen Reihenfolge die Tasks abgearbeitet werden.

Für jede Task kann eine Zeitüberwachung (Watchdog) konfiguriert werden.

Zusätzlich können Sie Systemereignisse (z. B. Start, Stop, Reset) direkt mit der Ausführung eines Projektbausteins koppeln.

Die Taskkonfiguration ist ein Objekt in der Registerkarte [Ressourcen] ([Resources]) des **Object Organizer**.

Der Task-Editor besteht aus einem zweigeteilten Fenster.

- Im linken Bereich werden die Tasks in einem Konfigurationsbaum dargestellt.

  In der ersten Zeile steht das Element [Taskkonfiguration] ([Task configuration]).

  Darunter folgt das Element [System-Ereignisse] ([System events]).

  Darunter folgen die Elemente der einzelnen Tasks, die durch den Tasknamen repräsentiert werden.

  Unterhalb jedes Taskelements stehen die zugehörigen Programmaufrufe.
- Im rechten Brereich befindet sich die Registerkarte mit den Eigenschaften des markierten Elements. Hier können Sie die einzelnen Tasks, Programmaufrufe bzw. Systemereignisse definieren.

Hinweis:

Verwenden Sie nicht in mehreren Tasks gleiche String-Funktionen, da sonst bei der Abarbeitung der Tasks die Gefahr des Überschreibens besteht.

6.6.2 [Einfügen] / [Task einfügen] oder [Einfügen] / [Task anhängen]

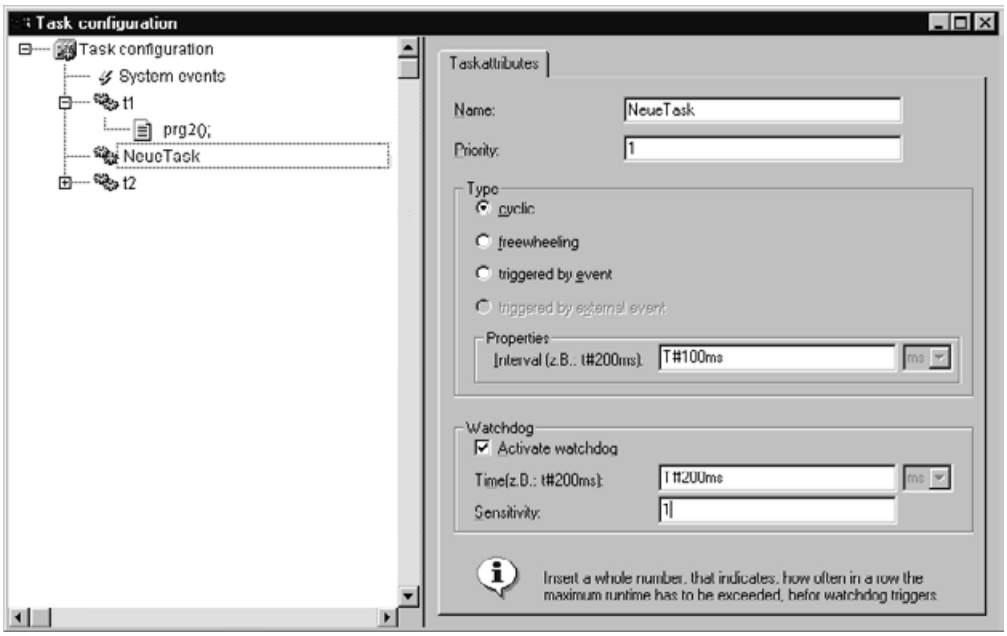
Mit dem Menübefehl [Einfügen] / [Task einfügen] ([Insert] / [Insert Task]) oder [Einfügen] / [Task anhängen] ([Insert] / [Append Task]) fügen Sie der Taskkonfiguration eine neue Task hinzu. Die Einträge bestehen jeweils aus einem Symbol und dem Tasknamen.

Der Menübefehl [Task einfügen] ([Insert Task]) steht nur zur Verfügung, wenn ein Task-Element oder das Element [System-Ereignisse] ([System events]) ausgewählt ist. Die neue Task wird unter der ausgewählten Task eingefügt.

Der Menübefehl [Task anhängen] ([Append Task]) steht zur Verfügung, wenn das Element [Taskkonfiguration] ([Task configuration]) ausgewählt ist. Die neue Task wird am Ende der bestehenden Liste angehängt.

Die maximal mögliche Anzahl von Tasks hängt von der Steuerung ab.

Wenn Sie eine Task einfügen, erscheint die Registerkarte [Taskeigenschaften] ([Taskattributes]), in der Sie die Taskeigenschaften festlegen.



20535AEN

Geben Sie die folgenden Attribute ein:

[Name]	[Name]	Geben Sie den Namen der Task ein, der im Konfigurationsbaum erscheint. Den Namen können Sie auch im Konfigurationsbaum ändern, indem Sie den Namen anklicken oder die <Leer>-Taste drücken und den Namen im Eingabefeld ändern.
[Priorität]	[Priority]	Geben Sie eine Zahl zwischen 0 und 31 ein, wobei 0 die höchste und 31 die niedrigste Priorität darstellt.
[Typ]	[Type]	Klicken Sie den gewünschten Typ an. • [Zyklisch] ([cyclic]): Die Task ist eine zyklische Task. Die Task wird entsprechend der im Feld [Eigenschaften] ([Properties]) eingegebenen Zeit zyklisch gestartet • [Freilaufend] ([Freewheeling]): Die Task wird beim Programmstart gestartet und nach jeder Abarbeitung neu gestartet. Es gibt keine Zykluszeitvorgaben. • [Extern ereignisgesteuert] ([triggered by external event]): Die Task ist ereignisgesteuert. Die Task wird entsprechend der im Feld [Ereignis] ([Event]) eingegebenen Zeit zyklisch gestartet.
[Watchdog]	[Watchdog]	In diesem Bereich können Sie für jede Task eine Zeitüberwachung konfigurieren. Wenn die Option aktiv ist (☑), erkennt die Steuerung einen Fehlerstatus, wenn sie bei der Abarbeitung die Watchdog-Zeit überschreitet (Watchdog-Mechanismus).
[Zeit (z.B.: t#200ms)]	[Time (e.g. t#200ms)]	Nach Ablauf dieser Watchdog-Zeit wird der Watchdog-Mechanismus aktiviert, falls die Task nicht selbstständig beendet wurde.
[Empfindlichkeit]	[Sensitivity]	Anzahl der Überschreitungen der Watchdog-Zeit, die akzeptiert wird, ohne dass die Steuerung in den Fehlerzustand versetzt wird.

Hinweis:

In Verbindung mit der Steuerungskarte Typ DHP11B ist die Empfindlichkeit ein Multiplikator. Die Watchdog-Zeit, nach der der Watchdog-Mechanismus auslöst, ergibt sich aus: [Zeit] ([Time]) * [Empfindlichkeit] ([Sensitivity]).

In Verbindung mit der Steuerungskarte Typ DHP11B werden zyklische Tasks über den Typ [Extern ereignisgesteuert] eingestellt. Werden hierbei mehrere Tasks mit der gleichen Zykluszeit parametriert, dann wird von den Tasks mit der gleichen Zykluszeit nur die letzte Task ausgeführt.

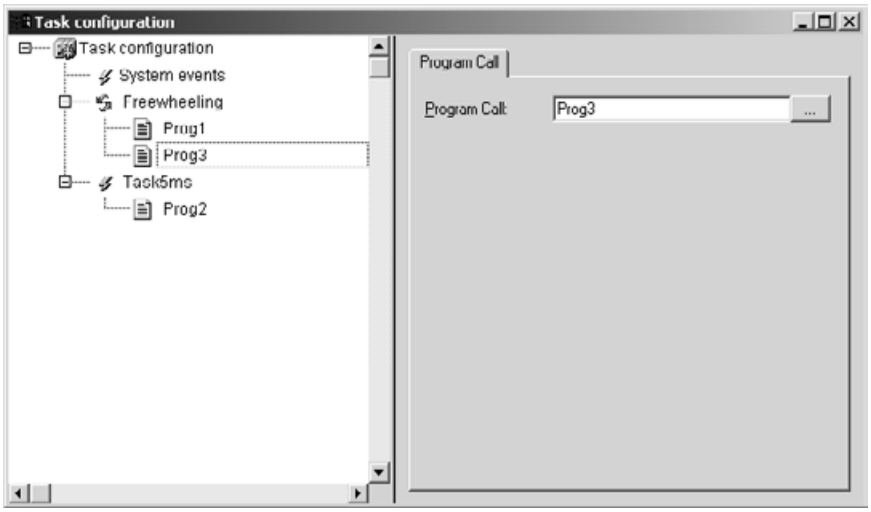
6.6.3 [Einfügen] / [Programmaufruf einfügen] oder [Einfügen] / [Programmaufruf anhängen]

Mit den Menübefehlen [Einfügen] / [Programmaufruf einfügen] ([Insert] / [Insert Program Call]) und [Einfügen] / [Programmaufruf anhängen] ([Insert] / [Append Program Call]) fügen Sie in der Taskkonfiguration zu einer Task einen Programmaufruf hinzu. Das Element im Konfigurationsbaum besteht aus dem Symbol und dem Programmnamen.

Mit dem Menübefehl [Programmaufruf einfügen] ([Insert Program Call]) fügen Sie einen neuen Programmaufruf vor dem ausgewählten Programmaufruf ein.

Mit dem Menübefehl [Programmaufruf anhängen] ([Append Program Call]) fügen Sie einen neuen Programmaufruf am Ende der bestehenden Liste der Programmeinträge ein.

Wenn Sie einen dieser Befehle ausführen wird die folgende Registerkarte angezeigt:



20536AEN

Geben Sie in das Feld [Programmaufruf] ([Program Call]) einen gültigen Programmnamen aus Ihrem Projekt ein. Alternativ können Sie auch mit der Schaltfläche [...] oder der <F2>-Taste die Eingabehilfe zur Auswahl gültiger Programmnamen öffnen und ein Programm auswählen.

Wenn der Programmeintrag ausgewählt ist, können Sie den Programmnamen auch im Konfigurationsbaum verändern. Öffnen Sie dazu das Editierfeld, indem Sie auf den Namen klicken oder die <Leer>-Taste drücken und anschließend den Namen ändern.

Wenn das ausgewählte Programm Eingabevariablen erfordert, geben Sie diese in der üblichen Form und gemäß dem deklarierten Typ an, z .B. `prg(invar := 17`.

Die Abarbeitung der Programmaufrufe erfolgt im Online-Modus gemäß der Reihenfolge ihrer Anordnung von oben nach unten.

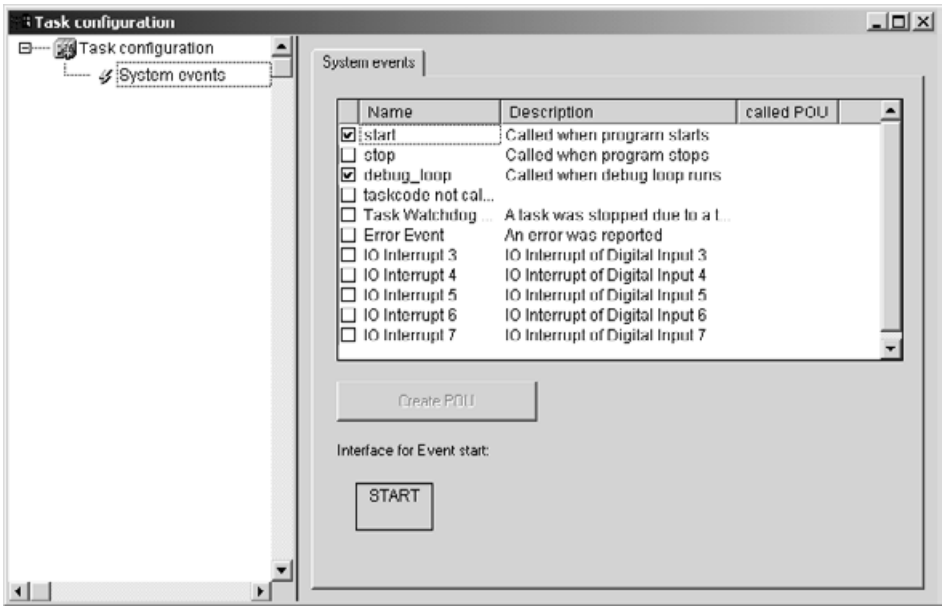
Hinweis:

Verwenden Sie nicht in mehreren Tasks gleiche String-Funktionen, da sonst bei der Abarbeitung der Tasks die Gefahr des Überschreibens besteht.

6.6.4 System-Ereignisse

Anstelle einer Task kann auch ein System-Ereignis (Event) einen Projektbaustein zur Abarbeitung aufrufen. Mögliche Ereignisse sind z. B. Stop, Start, Error Event.

Die Zuordnung der System-Ereignisse zu dem jeweils aufzurufenden Baustein erfolgt in der Registerkarte [System-Ereignisse] ([System events]). Diese Registerkarte erscheint, wenn im Konfigurationsbaum der Eintrag [System-Ereignisse] ([System events]) ausgewählt ist:



20537AEN

Jedes Ereignis wird in einer Tabellenzeile dargestellt:

Die Werte in den Spalten [Name] ([Name]) und [Beschreibung] ([Description]) werden aus der Zielsystembeschreibung übernommen.

In der Spalte [aufgerufene POU] ([called POU]) geben Sie den Namen des Projektbausteins ein, der bei Eintreten des Ereignisses ausgeführt werden soll. Drücken Sie dazu die <F2>-Taste und wählen Sie einen existierenden Baustein aus oder geben Sie den Namen eines existierenden Bausteins ein (z. B. PLC_PRG).

Sie können auch einen neuen Baustein im Projekt anlegen, indem Sie einen neuen, noch nicht vorhandenen Namen eingeben und auf die Schaltfläche [Baustein erzeugen] ([Create POU]) klicken. Danach erscheint der neue Baustein in der Registerkarte [Bausteine] ([POUs]) des Object Organizer. Der neue Baustein enthält im Deklarationsteil automatisch die Definitionen der für das Ereignis eventuell nötigen Übergabeparameter. Diese unter Umständen erforderliche Parametrierung eines Ereignisses wird unterhalb der Zuordnungsliste als Baustein auch grafisch dargestellt, wenn der entsprechende Tabelleneintrag ausgewählt ist.

Der Aufruf eines Bausteins durch das Ereignis erfolgt nur, wenn der Eintrag aktiv ist, d.h. wenn das Kontrollkästchen in der ersten Spalte einen Haken enthält [☑].

Folgende System-Ereignisse können Sie konfigurieren:

[Start]	Das Programm wird einmalig beim Start der Steuerung MOVI-PLC® bearbeitet.
[Stop]	Das Programm wird einmalig beim Stop der Steuerung MOVI-PLC® bearbeitet.
[Debug_Loop]	Das Programm wird bearbeitet, solange eine Task der Steuerung MOVI-PLC® an einem Breakpoint steht.
[Taskcode not called]	Das Programm wird bearbeitet, wenn auf der Steuerung MOVI-PLC® ein Programm geladen ist und die Steuerung MOVI-PLC® sich im Zustand Stop befindet.
[Task Watchdog Violation]	Das Programm wird einmal bearbeitet, wenn eine Task durch einen Watchdog-Mechanismus gestoppt wurde.
[Error Event]	Das Programm wird bearbeitet, wenn ein Baustein einer SEW-Bibliothek einen Fehler hervorgerufen hat.
[IO Interrupt 3-7]	Das Programm wird einmal bearbeitet, wenn ein Interruptereignis an einem Eingang der Steuerung MOVI-PLC® eingetreten ist.

Hinweis:

Bei Einsatz der Steuerungskarte DHP11B dürfen die Bausteine, die bei Eintreten von IO Interrupts ausgeführt werden, nur sehr wenige Anweisungen beinhalten. Aufwändige Programmteile, wie z.B. Buszugriffe oder umfangreiche Schleifen, müssen aus der Interrupttask heraus signalisiert bzw. angetriggert und in einer Task mit niedrigerer Priorität bearbeitet werden. Die Interrupttask hat eine höhere Priorität als die anderen System-Tasks.

6.6.5 Welche Task wird bearbeitet?

Bei der Ausführung der Tasks gelten folgende Regeln:

- Eine Task wird ausgeführt, wenn deren Bedingung gültig ist:
  - Wenn die im Feld [Zeit] ([Time]) angegebene Zeit abgelaufen ist
  - Wenn in der Bedingungsvariable eine steigende Flanke aufgetreten ist. Die Bedingungsvariable ist im Feld [Ereignis] ([Event]) angegeben.
- Wenn die Bedingungen mehrerer Tasks gültig sind, wird die Task mit der höchsten Priorität ausgeführt.
- Wenn die Bedingungen mehrerer Tasks gültig sind und Sie die gleiche Priorität haben, wird die Task ausgeführt, die die längste Wartezeit hatte.
- Die Abarbeitung der Programmaufrufe pro Task erfolgt im Online-Modus gemäß der Reihenfolge ihrer Anordnung im Task-Editor von oben nach unten.
- Wenn das Programm mit dem Namen `PLC_PRG` eingefügt wird, dann läuft dieses automatisch in der freilaufenden Task. Ein Eintrag in der Taskkonfiguration ist nicht nötig.

Hinweis:

Bei Einsatz der Steuerungskarte Typ DHP11B hat der Parameter [Priorität] ([Priority]) keine Funktion.

Die zyklische Task mit der kürzesten Zykluszeit hat die höchste Priorität. Die freilaufende Task hat immer die niedrigste Priorität.

Bei Einsatz der Steuerungskarte Typ DHP11B hat der Parameter [Priorität]([Priority)] keine Funktion. Die Systemereignisse (siehe Kap. 6.6.4) haben die höchste Priorität, gefolgt von den zyklischen Tasks. Je kürzer die Zykluszeit ist, desto höher die Priorität. Die freilaufende Task hat die niedrigste Priorität.

6.6.6 [Extras] / [Debug Task festlegen]

Mit dem Menübefehl [Extras] / [Debug Task festlegen] ([Extras] / [Debug Task festlegen] legen Sie bei Zielsystemen mit "preemptive multitasking" im Online-Modus in der Taskkonfiguration eine Debugging-Task fest. Im Taskkonfigurationsbaum erscheint dann hinter dem Taskeintrag der Text DEBUG. Die Debugging-Funktionalitäten beziehen sich dann nur auf die ausgewählte Task. Das Programm stoppt bei einem Breakpoint nur, wenn das Programm von der eingestellten Task durchlaufen wird.

Die Festlegung der Debugging-Task wird im Projekt gespeichert und bei Einloggen/Download automatisch wieder gesetzt.

Hinweis:

Die Steuerungskarte Typ DHP11B unterstützt den Befehl [Extras] / [Debug Task festlegen] ([Extras] / [Debug Task festlegen] nicht. Wenn in einem Programm ein Breakpoint gesetzt ist, werden alle angelegten Tasks gestoppt.

6.6.7 [Extras] / [Task aus-/einschalten]

Mit dem Menübefehl [Extras] / [Task aus-/einschalten] ([Extras] / [Enable/disable Task]) schalten Sie die in der Taskkonfiguration ausgewählte Task aus oder wieder ein. Wenn eine Task ausgeschaltet ist, wird sie in der Abarbeitung des Programms nicht berücksichtigt. Wenn die Task inaktiv ist, wird sie im Konfigurationsbaum in hellgrauer Schrift angezeigt.

6.6.8 [Extras] / [Aufrufhierarchie]

Wenn das Programm beim **Debugging** an einem **Breakpoint** stoppt, können Sie mit dem Menübefehl [Extras] / [Aufrufhierarchie] ([Extras] / [Callstack]) die Aufrufhierarchie des betreffenden Bausteins ermitteln. Dazu muss die Debugging-Task im Konfigurationsbaum der Taskkonfiguration ausgewählt sein. Der Dialog [Aufrufhierarchie von Task] / [Callstack from Task] wird geöffnet. Der Dialog zeigt den Namen des Bausteins an, in dem der Breakpoint liegt an, z. B. `prog_x (2)` für Zeile 2 des Bausteins `prog_x`. Danach folgen in rücklaufender Reihenfolge die Einträge für die aufrufenden Bausteinpositionen. Wenn Sie auf die Schaltfläche [Gehe zu] ([Go To]) klicken, springt der Fokus zur ausgewählten Position.

Hinweis:

Die Steuerungskarte Typ DHP11B unterstützt den Menübefehl [Extras] / [Aufrufhierarchie] ([Extras] / [Callstack]) nicht. Wenn in einem Programm ein Breakpoint gesetzt ist, werden alle angelegten Tasks gestoppt.

6.7 Traceaufzeichnung

6.7.1 Überblick und Konfiguration

Traceaufzeichnung

Während einer Traceaufzeichnung wird der Werteverlauf von Variablen über einen bestimmten Zeitraum hin aufgezeichnet. Diese Werte werden in einem Ringspeicher gespeichert (Tracebuffer). Wenn der Speicher voll ist, werden die ältesten Werte vom Speicheranfang her wieder überschrieben.

Maximal 20 Variablen können gleichzeitig aufgezeichnet werden. Pro Variable können maximal 500 Werte aufgezeichnet werden. Da die Größe des Tracebuffer in der Steuerung einen festen Wert besitzt, können bei sehr vielen oder sehr breiten Variablen (`DWORD`) weniger als 500 Werte aufgezeichnet werden.

Beispiel:

Wenn zehn Variablen von Typ `WORD` aufgezeichnet werden sollen und der Speicher in der Steuerung 5000 Byte lang ist, können von jeder Variablen 250 Werte aufgezeichnet werden.

Hinweis:

Bei der Steuerungskarte Typ MOVI-PLC® DHP11B ist der Tracebuffer 5000 Byte lang.

Um einen Trace aufzeichnen zu können, öffnen Sie das Objekt [Traceaufzeichnung] ([Sampling Trace]) in der Registerkarte [Ressourcen] ([Resources]) des Object Organizer. Erstellen bzw. laden Sie eine entsprechende Tracekonfiguration und definieren Sie die Tracevariablen, die aufgezeichnet werden sollen.

Weitere Informationen finden Sie auf Seite 311 (→ Abschnitt. [Extras] / [Tracekonfiguration]) und auf Seite 313 (→ Abschnitt. Auswahl der darzustellenden Variablen).

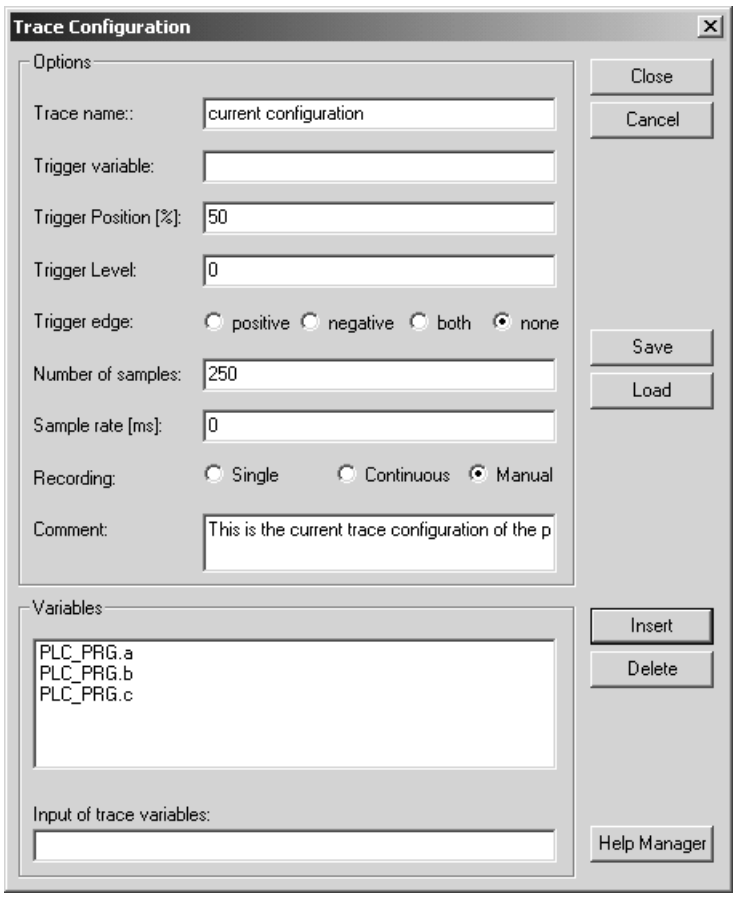
Nachdem Sie die Konfiguration im Tracekonfigurationsdialog erstellt und die Aufzeichnung in der Steuerung [Trace starten] ([Start Trace]) gestartet haben, werden die Werte der Variablen aufgezeichnet. Mit dem Menübefehl [Trace lesen] ([Read Trace]) werden die zuletzt aufgezeichneten Werte ausgelesen und grafisch als Kurven dargestellt.

Die Traceaufzeichnung (Variablenwerte und Konfiguration) können Sie im Projektformat `*.trc` oder im XML-Format `*.mon` speichern und wieder laden. Die Konfiguration alleine können Sie in einer `*.tcf`-Datei speichern und wieder laden.

Im Projekt stehen verschiedene Aufzeichnungen zur Ansicht zur Verfügung. Die verfügbaren Aufzeichnungen können Sie in der Auswahlliste [Trace] in der rechten oberen Ecke des Dialogs [Traceaufzeichnung] ([Sampling Trace]) auswählen. In der Auswahlliste können Sie die zu verwendende Traceaufzeichnung auswählen.

[Extras] / [Tracekonfiguration]

Nach dem Ausführen des Menübefehls [Extras] / [Tracekonfiguration] ([Extras] / [Trace configuration]) wird der Dialog [Trace Konfiguration] ([Trace Configuration]) geöffnet. In diesem Dialog wählen Sie die aufzuzeichnenden Variablen aus und geben die Parameter der Traceaufzeichnung ein. Den Dialog können Sie ebenso öffnen, indem Sie im Dialog [Sampling Trace] ([Sampling Trace]) auf die graue Fläche doppelklicken.



20538AEN

Folgende Parameter können Sie eingeben:

[Trace Name]	[Trace name]	Geben Sie den Namen der Tracekonfiguration ein. Wenn Sie den Konfigurationsdialog mit der Schaltfläche [OK] verlassen, wird die Konfiguration in der Auswahlliste [Trace] des Dialogs [Traceaufzeichnung] ([Sampling Trace]) mit diesem Namen aufgelistet.
[Trigger Variable]	[Trigger variable]	In dieses Feld können Sie eine boolesche, analoge oder eine Enumerationsvariable eingeben. Sie können hier auch die Eingabehilfe verwenden, die Sie mit der <F2>-Taste öffnen. Die Trigger-Variable beschreibt die Abbruchbedingung des Trace. Wenn keine Trigger-Variable verwendet wird, können Sie die Traceaufzeichnung Mit dem Menübefehl [Extras] / [Trace stoppen] ([Extras] / [Stop Trace]) abbrechen.
[Trigger Position]	[Trigger Position]	In diesem Feld geben Sie ein, wieviel Prozent der Messwerte vor dem Eintreten des Triggerereignisses aufgezeichnet werden. **Beispiel:** Wenn Sie 25 eingeben, werden 25% der Messwerte vor und 75% der Messwerte nach dem Triggerereignis dargestellt. Anschließend wird der Trace abgebrochen.
[Tigger Level]	[Tigger Level]	In diesem Feld geben Sie an, welchen Wert eine analoge Trigger-Variable annehmen muss, damit das Triggerereignis eintritt. Sie können auch eine ENUM-Konstante eingeben, die den Trigger-Level festlegt.

[Trigger Flanke]	[Trigger edge]	Hier wählen Sie die Triggferflanke aus, die das Triggerereignis auslöst. • [positiv] ([positive]): Das Triggerereignis tritt nach einer steigenden Flanke einer booleschen Trigger-Variablen ein oder wenn eine analoge Trigger-Variable den Trigger-Level von unten nach oben überschreitet. • [negativ] ([negative]): Das Triggerereignis tritt nach einer fallenden Flanke einer booleschen Trigger-Variablen ein oder wenn eine analoge Trigger-Variable den Trigger-Level von oben nach unten überschreitet • [beide] ([both]): Das Triggerereignis tritt nach einer steigenden oder fallenden Flanke einer booleschen Trigger-Variablen ein oder wenn eine analoge Trigger-Variable den Trigger-Level überschreitet. • [keine] ([none]): Es gibt es kein Triggerereignis.
[Abtastrate]	[Sample rate]	In diesem Feld geben Sie die Zeit zwischen zwei Aufzeichnungen in Millisekunden ein. Die Vorbelegung 0 bedeutet, dass ein Abtastvorgang pro Zyklus stattfindet. **Hinweis:** Wenn eine Taskkonfiguration zum Steuern des Programmablaufs verwendet wird, bezieht sich die Trace-Funktion auf die Debug-Task. Für die Steuerungskarte Typ DHP11B beachten Sie die folgenden zulässigen Werte der Abtastrate: • Wenn nur die freilaufende Task verwendet wird, müssen Sie eine Abtastrate von 0 eingeben. Ein Zeitintervall kann nicht sinnvoll angegeben werden. • Wenn eine zyklische Task und keine freilaufende Task verwendet wird, dürfen Sie eine Abtastrate von 0, die Taskzykluszeit oder ein Vielfaches der Taskzykluszeit eingeben. Der Programmcode muss dabei innerhalb der Taskzykluszeit abgearbeitet sein. • Wenn eine freilaufende und eine oder mehrere Tasks oder keine freilaufende und eine oder mehrere Tasks verwendet werden, können Sie Werte der Task mit der längsten Zykluszeit nur korrekt aufzeichnen, wenn Sie eine Abtastrate von 0 eingeben. Ein Zeitintervall kann nicht sinnvoll angegeben werden.
[Aufzeichnung]	[Recording]	Hier wählen Sie den Aufruf-Modus der Traceaufzeichnung aus. • [Einzeln] ([Single]): Die vorgegebene Anzahl der Messungen wird einmal dargestellt. • [Fortlaufend] ([Continuous]): Das Auslesen der Aufzeichnung der vorgegebenen Messwert-Anzahl wird immer wieder neu gestartet. **Beispiel:** Wenn Sie im Feld [Anzahl Messungen] ([Number of samples]) 35 eingeben, umfasst die erste Darstellung die ersten Messwerte von 1 bis 35. Anschließend wird die Aufzeichnung der nächsten 35 Messwerte (36-70) automatisch dargestellt usw. • [Manuell] ([Manual]): Mit dem Menübefehl [Extras] / [Trace lesen] ([Extras] / [Read Trace]) lesen Sie die Traceaufzeichnung manuell aus. Der Abruf-Modus ist unabhängig davon, ob eine Trigger-Variable eingegeben ist oder nicht. Wenn keine Trigger-Variable eingegeben ist, wird der Tracebuffer mit der Anzahl der vorgegebenen Messwerte gefüllt. Beim Abruf wird der Bufferinhalt gelesen und dargestellt.
[Kommentar]	[Comment]	In diesem Feld können Sie einen beliebigen Kommentartext eingeben.
[Variablen]	[Variables]	Die Liste der aufzuzeichnenden Variablen ist zunächst leer. Um eine Variable einzufügen, haben Sie folgende Möglichkeiten: • Geben Sie den Namen der neuen Variablen in das Feld [Eingabe von Trace Variablen] ([Input of trace variables]) und klicken auf die Taste [Einfügen] ([Insert]), • Klicken Sie auf die Schaltfläche [Eingabehilfe] ([Help Manager]) und wählen Sie die Variable aus. Bestätigen Sie mit [OK]. Die Verwendung von Enumerationsvariablen ist möglich. Eine Variable löschen Sie aus der Variablen-Liste, indem Sie die Variable auswählen und auf die Schaltfläche [Löschen] ([Delete]) klicken.

Mit der Schaltfläche [Speichern] ([Save]) speichern Sie die erstellte Tracekonfiguration in einer Datei vom Format `*.tcf`. Der Dialog [Datei speichern unter] ([Save as]) wird geöffnet, in dem Sie den Dateinamen eingeben und den Pfad auswählen. Klicken Sie anschließend auf die Schaltfläche [Speichern] ([Save]).

Mit der Schaltfläche [Laden] ([Load]) laden Sie eine abgespeicherte Tracekonfiguration. Der Dialog [Öffnen] ([Open]) wird geöffnet, in dem Sie die gewünschte Tracekonfiguration und den Pfad auswählen. Klicken Sie anschließend auf die Schaltfläche [Öffnen] ([Open]).

Hinweis:

Wenn Sie eine Taskkonfiguration zum Steuern des Programmablaufs verwenden, bezieht sich die Tracefunktion auf die Debug-Task.

Beachten Sie bei Steuerungskarte MOVI-PLC® DHP11B folgende mögliche Angaben der Abtastrate:

- Wenn nur die freilaufende Task verwendet wird, muss im Feld [Abtastrate] der Wert "0" eingetragen werden.
- Wenn eine zyklische Task und keine freilaufende Task verwendet wird, kann als Abtastrate der Wert "0" oder die Taskzykluszeit oder ein Vielfaches der Taskzykluszeit eingetragen werden. Der Programmcode muss hierbei innerhalb der Taskzykluszeit abgearbeitet sein.
- Wenn mehrere Tasks verwendet werden, können nur bei einer Abtastrate "0" Werte der Task mit einer längeren Zykluszeit korrekt aufgezeichnet werden.

Auswahl der darzustellenden Variablen

Die Felder rechts neben dem Fenster für die Darstellung der Kurven im Dialog [Traceaufzeichnung] ([Sampling Trace]) enthalten alle in der Tracekonfiguration definierten Tracevariablen. Wenn Sie eine Variable aus der Liste auswählen, wird ein Tracebuffer gelesen und die Werte der Variablen werden in der entsprechenden Farbe dargestellt. Eine Variable können Sie nur auswählen, wenn deren Kurve ausgegeben wurde.

Im Tracefenster können maximal acht Variablen gleichzeitig dargestellt werden.

6.7.2 Eine Traceaufzeichnung durchführen

[Extras] / [Trace starten]

Schaltfläche: [ ]

Mit dem Menübefehl [Extras] / [Trace starten] ([Extras] / [Start Trace]) übertragen Sie die Tracekonfiguration in die Steuerung und starten die Traceaufzeichnung in der Steuerung.

[Extras] / [Trace lesen]

Schaltfläche: [ ]

Mit dem Menübefehl [Extras] / [Trace lesen] ([Extras] / [Read Trace]) lesen Sie den aktuellen Tracebuffer aus der Steuerung ein. Die Werte der ausgewählten Variablen werden dargestellt.

[Extras] / [Trace automatisch lesen]

Mit dem Menübefehl [Extras] / [Trace automatisch lesen] ([Extras] / [Auto Read Trace]) wird der aktuelle Tracebuffer aus der Steuerung automatisch gelesen und die Werte werden fortlaufend dargestellt.

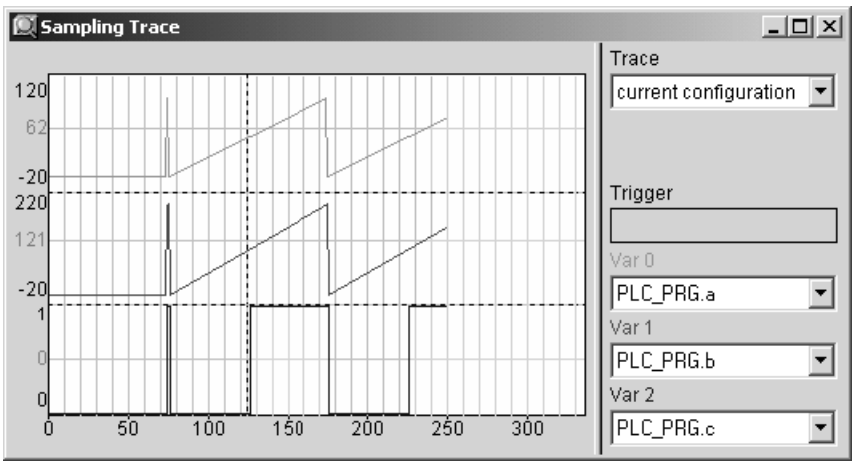
[Extras] / [Trace stoppen]

Schaltfläche: [ ]

Mit dem Menübefehl [Extras] / [Trace stoppen] ([Extras] / [Stop Trace]) beenden Sie die Traceaufzeichnung in der Steuerung.

6.7.3 Betrachten der Traceaufzeichnung

Darstellung der Traceaufzeichnung



20539AEN

Im Tracefenster wird rechts oben der Name der aktuell verwendeten Tracekonfiguration und rechts unten ein eventuell verfügbarer Kommentar angezeigt.

Wenn ein Tracebuffer geladen ist, können die Werte aller darzustellenden Variablen ausgelesen werden und werden im Tracefenster entsprechend dargestellt.

Wenn keine Abtastrate eingegeben wurde, wird die X-Achse mit der fortlaufenden Nummer des aufgezeichneten Werts beschriftet. Der Tracebuffer wird gelöscht, sobald die Aufzeichnung gestoppt wird.

Die Statusanzeige des Tracefensters zeigt an, ob der Tracebuffer noch nicht voll ist und ob die Traceaufzeichnung noch läuft oder bereits beendet ist.

Wenn ein Wert für die Abtastrate eingegeben wurde, zeigt die X-Achse die Zeit des Messwerts an. Dem ältesten aufgezeichneten Messwert wird die Zeit 0 zugeordnet. Das Beispiel zeigt die Werte innerhalb der letzten 250 ms an.

Die Y-Achse wird mit den Werten des entsprechenden Datentyps beschriftet. Die Skalierung ist so ausgelegt, dass der niedrigste und der höchste Wert in den Bildbereich passen. Im Beispiel hat `Var0` den niedrigsten Wert 0 und den höchsten Wert 100. Daraus folgt die Einstellung der Skala am linken Rand.

Wenn die Triggerbedingung erfüllt ist, wird die Stelle zwischen den Werten vor und nach dem Eintreten der Triggerbedingung durch eine senkrechte gestrichelte Linie dargestellt.

[Extras] / [Cursor ausgeben]

Am einfachsten setzen Sie den Cursor im Tracefenster, indem Sie mit der linken Maustaste auf das Tracefenster klicken. Mit der Maus können Sie den Cursor beliebig verschieben. Über dem Graphikfenster können Sie jeweils die aktuelle x-Position des Cursors ablesen. Neben den Feldern [Var0], [Var1], ...,[VarN] wird der Wert der jeweiligen Variable angezeigt.

Mit dem Menübefehl [Extras] / [Cursor ausgeben] ([Extras] / [Cursor Mode]) fügen Sie zwei vertikale Linien in die Traceaufzeichnung ein, die zunächst übereinander liegen. Verschieben Sie eine der Linien mit den Pfeiltasten nach rechts und links.

Um die Geschwindigkeit der Bewegung um den Faktor 10 zu erhöhen, drücken Sie die Tasten <Strg>+<Pfeil nach links> bzw. <Strg>+<Pfeil nach rechts>-Tasten, .

Wenn Sie zusätzlich die <Umschalt>-Taste drücken, verschieben Sie die andere Linie, die den Differenzbetrag zur ersten Linie anzeigt.

[Extras] / [Mehrkanal]

Mit dem Menübefehl [Extras] / [Mehrkanal] ([Extras] / [Multi Channel]) wechseln Sie zwischen einkanaliger und mehrkanaliger Darstellung der Traceaufzeichnung. Bei mehrkanaliger Darstellung erscheint ein Haken vor dem Menübefehl.

Voreingestellt ist die mehrkanalige Darstellung. Dabei wird das Darstellungsfenster auf die bis zu acht darzustellenden Kurven aufgeteilt. Am Rand wird der maximale und der minimale Wert jeder Kurve dargestellt.

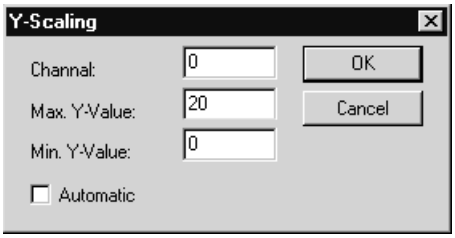
Bei einkanaliger Darstellung werden alle Kurven mit dem gleichen Skalierungsfaktor dargestellt und überlagert. Diese Darstellung ist nützlich, um Abweichungen von Kurven darzustellen.

[Extras] / [Koordinatennetz anzeigen]

Mit dem Menübefehl [Extras] / [Koordinatennetz anzeigen] ([Extras] / [Show grid]) schalten Sie das Koordinatennetz im Darstellungsfenster der Traceaufzeichnung ein oder aus. Wenn das Koordinatennetz eingeschaltet ist, erscheint ein Haken vor dem Menübefehl.

[Extras] / [Y-Skalierung]

Mit dem Menübefehl [Extras] / [Y-Skalierung] ([Extras] / [Y scaling]) ändern Sie die vorgegebene Y-Skalierung einer Kurve in der Tracedarstellung. Der Dialog [Y-Skalierung] ([Y Scaling]) wird geöffnet, in dem Sie die Y-Skalierung ändern. Wenn Sie auf eine Kurve doppelklicken, erscheint dieser Dialog ebenfalls.



20540AEN

Wenn die Option [Automatisch] ([Automatic]) aktiviert ist, wird die Default-Skalierung verwendet, die vom Typ der entsprechenden Variable abhängt. Bei Enumerationen werden die entsprechenden Enumerationswerte als Skalenbeschriftung angezeigt.

Um die Skalierung zu verändern, deaktivieren Sie die Option [Automatisch] ([Automatic]) und geben Sie folgende Werte ein:

[Kanal]	[Channel]	Geben Sie die Nummer der gewünschten Kurve ein.
[Max.Y-Wert]	[Max.Y-Value]	Geben Sie den neuen höchsten Y-Wert ein.
[Min.Y-Wert]	[Min.Y-Value]	Geben Sie den neuen niedrigsten Y-Wert ein.

Schließen Sie den Dialog mit [OK].

[Extras] / [Strecken]

Schaltfläche: [ ]

Mit dem Menübefehl [Extras] / [Strecken] ([Extras] / [Stretch]) strecken (zoomen) Sie die Werte der Traceaufzeichnung. Die Anfangsposition stellen Sie mit der horizontalen Bildlaufleiste ein. Wenn Sie diesen Menübefehl mehrmals nacheinander ausführen, wird ein immer kürzerer Traceausschnitt im Fenster angezeigt.

Die entgegengesetzte Wirkung erzielen Sie mit dem Menübefehl [Extras] / [Komprimieren] ([Extras] / [Compress]).

[Extras] / [Komprimieren]

Schaltfläche: [ ]

Mit dem Menübefehl [Extras] / [Komprimieren] ([Extras] / [Compress]) komprimieren Sie die Werte der Traceaufzeichnung. Wenn Sie diesen Menübefehl mehrmals nacheinander ausführen, wird ein immer größerer Traceausschnitt im Fenster angezeigt.

Die entgegengesetzte Wirkung erzielen Sie mit dem Menübefehl [Extras] / [Strecken] ([Extras] / [Stretch]).

6.7.4 [Extras] / [Tracewerte speichern]

Übersicht

Die Menübefehle des Menüpunkts [Extras] / [Tracewerte speichern] ([Extras] / [Save trace values]) dienen dazu, die Werte und die Tracekonfiguration einer Traceaufzeichnung in einer Datei im Projektformat zu speichern oder aus einer solchen Datei zu laden. Außerdem können Sie die Aufzeichnung in einer ASCII-Datei speichern.

Hinweis:

Beachten Sie die alternativen Speicher- und Ladebefehle des Menüpunkts [Extras] / [Externe Tracekonfigurationen] ([Extras] / [External trace configuration]).

[Werte speichern]

Mit dem Menübefehl [Extras] / [Tracewerte speichern] / [Werte speichern] ([Extras] / [Save trace values] / [Save values]) speichern Sie die Werte und die Konfiguration der Traceaufzeichnung. Der Dialog [Datei speichern unter] ([Save as]) wird geöffnet, in dem Sie den Dateinamen eingeben und den Pfad auswählen. Der Dateiname erhält die Erweiterung `.trc`. Klicken Sie anschließend auf die Schaltfläche [Speichern] ([Save]).

Die gespeicherte Traceaufzeichnung können Sie mit dem Menübefehl [Werte laden] ([Load values] wieder laden.

[Werte laden]

Mit dem Menübefehl [Extras] / [Tracewerte speichern] / [Werte laden] ([Extras] / [Save trace values] / [Load values]) laden Sie eine abgespeicherte Traceaufzeichnung (Werte und Konfiguration) wieder. Der Dialog [Datei öffnen] ([Open file]) wird geöffnet. Wählen Sie die gewünschte Datei mit der Erweiterung `.trc` aus. Die Aufzeichnung wird im Tracefenster dargestellt und die Konfiguration wird als aktuelle Konfiguration im Projekt übernommen.

Die Traceaufzeichnung können Sie mit dem Befehl [Werte speichern] ([Save values]) in einer `*.trc`-Datei speichern.

[Werte in ASCII-File]

Mit dem Menübefehl [Extras] / [Tracewerte speichern] / [Werte in ASCII-File] ([Extras] / [Save trace values] / [Values in ASCII-File]) speichern Sie die Werte einer Traceaufzeichnung in einer ASCII-Datei ab. Der Dialog [Datei speichern unter] ([Save as]) wird geöffnet, in dem Sie den Dateinamen eingeben und den Pfad auswählen. Der Dateiname erhält die Erweiterung `.txt`. In der Datei werden die Werte gemäß folgendem Schema abgespeichert:

```
PLCEditor Trace
D:\PLCEditor\PROJECTS\AMPEL.PRO
Zyklus PLC_PRG.ZAEHLER PLC_PRG.LIGHT1
0 2 1
1 2 1
2 2 1
.....
```

Wenn in der Tracekonfiguration keine Abtastrate eingegeben wurde, steht in der ersten Spalte der Zyklus. Pro Zyklus wurde also ein Wert erfasst.

Wenn eine Abtastrate eingegeben wurde, steht in der ersten Spalte der Zeitpunkt in

[ms], an dem die Werte der Variablen gespeichert wurden. Die Zeitrechnung startet mit der Traceaufzeichnung.

In den darauf folgenden Spalten werden die entsprechenden Werte der Tracevariablen abgespeichert. Die Werte sind jeweils durch ein Leerzeichen voneinander getrennt.

In der dritten Zeile werden die zugehörigen Variablennamen der Reihenfolge nach nebeneinander dargestellt (`PLC_PRG.ZAEHLER, PLC_PRG.LIGHT1`).

6.7.5 [Extras] / [Externe Tracekonfiguration]

Übersicht

Die Menübefehle des Menüpunkts [Extras] / [Externe Tracekonfiguration] ([Extras] / [External trace configuration]) dienen dazu, die Werte und die Tracekonfiguration einer Traceaufzeichnung in Dateien zu speichern bzw. die Werte aus Dateien oder von der Steuerung ins Projekt zu laden. Außerdem können Sie die im Projekt zu verwendende Konfiguration festgelegen.

Hinweis:

Beachten Sie auch die Speicher- und Ladebefehle des Menüs [Extras] / [Tracewerte speichern] ([Extras] / [Save trace values]).

[Speichern in Datei]

Mit dem Menübefehl [Extras] / [Externe Tracekonfiguration] / [Speichern in Datei] ([Extras] / [External trace configuration] / [Save to file]) speichern Sie eine Traceaufzeichnung (Werte und Konfiguration) in einer Datei im XML-Format. Der Dialog [Datei speichern unter] ([Save as]) wird geöffnet, in dem Sie den Dateinamen eingeben und den Pfad auswählen. Der Dateiname erhält die Erweiterung `.mon`. Klicken Sie anschließend auf die Schaltfläche [Speichern] ([Save]).

Eine `*.mon`-Datei können Sie mit dem Menübefehl [Laden von Datei] ([Load from file]) in das Projekt laden.

Hinweis:

Beachten Sie, dass hier sowohl die Werte als auch die Tracekonfiguration im XML-Format gespeichert werden. Beim Speichern im Konfigurationsdialog wird nur die Konfiguration gespeichert.

[Laden von Datei]

Mit dem Menübefehl [Extras] / [Externe Tracekonfiguration] / [Laden von Datei] ([Extras] / [External trace configuration] / [Load from file]) laden Sie die Werte und die Konfiguration einer Traceaufzeichnung, die in einer Datei im XML-Format vorliegt, in das aktuelle Projekt. Der Dialog [Öffnen] ([Open]) wird geöffnet. Wählen Sie die gewünschte Datei mit der Erweiterung `.mon` aus. Die neu geladene Traceaufzeichnung wird im Tracefenster dargestellt. Die neu geladene Konfiguration wird in der Auswahlliste im Feld [Trace] hinzugefügt. Um die neu geladene Konfiguration zur aktuellen Projektkonfiguration zu machen, führen Sie den Menübefehl [Extras] / [Als Projektkonfiguration übernehmen] ([Extras] / [Apply as project configuration]) aus.

Eine `*.mon`-Datei können Sie mit dem Menübefehl [Speichern in Datei] ([Save to file]) erzeugen.

[Speichern auf Steuerung]

Mit dem Menübefehl [Extras] / [Externe Tracekonfiguration] / [Speichern auf Steuerung] ([Extras] / [External trace configuration] / [Save to target]) laden Sie im Online-Modus eine Traceaufzeichnung, die in einer Datei im XML-Format vorliegt, in die Steuerung. Der Dialog [Datei speichern unter] ([Write file to PLC]) wird geöffnet, in dem Sie den Dateinamen und den Pfad auswählen. Zunächst werden die Dateien mit der Erweiterung `.mon` angezeigt.

Beachten Sie, dass Sie die Tracekonfiguration im XML-Format in `*.mon`-Dateien speichern können. Weitere Informationen finden Sie auf Seite 318 (→ Abschnitt. [Speichern in Datei]) und auf Seite 319 (→ Abschnitt. [Laden von Steuerung]).

Hinweis:

Die Steuerungskarte Typ DHP11B unterstützt diese Funktion nicht.

[Laden von Steuerung]

Mit dem Menübefehl [Extras] / [Externe Tracekonfiguration] / [Laden von Steuerung] ([Extras] / [External trace configuration] / [Load from target]) laden Sie die aktuelle Traceaufzeichnung der Steuerung in das Projekt. Dabei werden die Konfiguration und die Werte aus einer Datei im XML-Format geladen. Die Traceaufzeichnung wird im Tracefenster dargestellt. Die geladene Tracekonfiguration können Sie als aktuelle Projektkonfiguration übernehmen.

Weitere Informationen finden Sie auf Seite 319 (→ Abschnitt. [Speichern auf Steuerung]).

Hinweis:

Die Steuerungskarte Typ DHP11B unterstützt diese Funktion nicht.

[Als Projektkonfiguration übernehmen]

Mit dem Menübefehl [Extras] / [Externe Tracekonfiguration] / [Als Projektkonfiguration übernehmen] ([Extras] / [External trace configuration] / [Apply as project configuration]) wird die Tracekonfiguration, die in der Auswahlliste [Trace] des Konfigurationsdialogs ausgewählt ist, als aktuell aktive Tracekonfiguration im Projekt übernommen.

Die Auswahlliste bietet neben der momentan aktiven Konfiguration alle Konfigurationen an, die mit dem Menübefehl [Laden von Datei] ([Load from file]) aus den `*.mon`-Dateien bereits ins Projekt geladen wurden.

6.8 Watch- und Rezepturverwalter

6.8.1 Übersicht

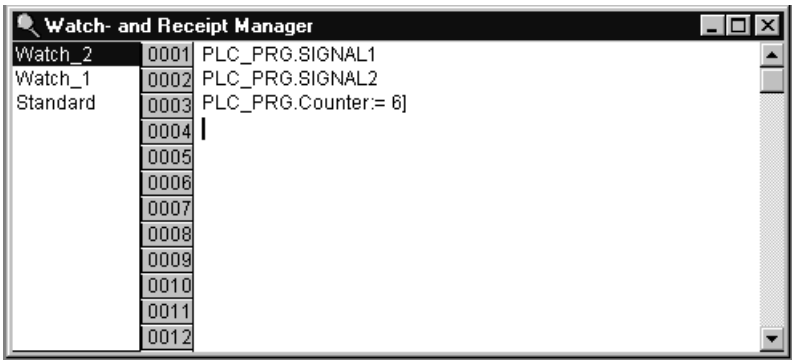
Mit Hilfe des Watch- und Rezepturverwalters können Sie sich die Werte von ausgewählten Variablen anzeigen lassen. Außerdem können Sie im Watch- und Rezepturverwalter Variablen mit bestimmten Werten vorbelegen und mit dem Menübefehl [Rezeptur schreiben] ([Write recipe]) auf einmal an die Steuerung übertragen. Außerdem können Sie die aktuellen Werte der Steuerung als Vorbelegung in den Watch- und Rezepturverwalter einlesen und abgespeichern. Besonders hilfreich sind diese Funktionen z. B. für die Einstellung und Erfassung von Regelungsparametern.

Alle erzeugten Watchlisten werden in der linken Spalte des Watch- und Rezepturverwalter angezeigt und können mit einem Mausklick oder den Pfeiltasten ausgewählt werden. Im rechten Bereich des Watch- und Rezepturverwalters werden die jeweils zugehörigen Variablen angezeigt.

Um mit dem Watch- und Rezepturverwalter zu arbeiten, öffnen Sie das Objekt [Watch- und Rezepturverwalter] ([Watch- and Receipt Manager]) in der Registerkarte [Ressourcen] ([Resources]) des Object Organizers.

6.8.2 Watch- und Rezepturverwalter im Offline-Modus

Im Offline-Modus können Sie mit dem Menübefehl [Einfügen] / [Neue Watchliste] ([Insert] / [New Watch List] im Watch- und Rezepturverwalter mehrere Watchlisten erzeugen.



20541AEN

Um die zu beobachtenden Variablen auszuwählen, rufen Sie die Liste aller Variablen mit der <F2>-Taste auf oder geben Sie die Variablen mit der Tastatur gemäß folgender Notation ein:

`<Bausteinname>.<Variablenname>`

Bei globalen Variablen fehlt der Bausteinname. Globale Variablen beginnen mit einem Punkt. Der Variablenname kann wiederum mehrstufig sein. Adressen können Sie direkt eingeben.

Beispiel

Mehrstufige Variable

```
PLC_PRG.Instanz1.Instanz2.Struktur.Komponentenname
```

Globale Variable:

```
.global1.component1
```

Die Variablen der Watchliste können Sie mit konstanten Werten vorbelegen. Im Online-

Modus können Sie diese Werte mit dem Menübefehl [Extras] / [Rezeptur schreiben] ([Extras] / [Write recipe]) in die Variablen schreiben.

Dazu müssen Sie der Variablen einen konstanten Wert mit der Zuweisung `:=` zuweisen:

Beispiel

```
PLC_PRG.TIMER:=50
```

In diesem Beispiel ist Variable `PLC_PRG.TIMER` mit 50 vorbelegt. In der vorherigen Abbildung ist die Variable `PLC_PRG.Counter` mit dem Wert 6 vorbelegt.

Um Variablen vom Typ Array, Struktur oder Funktionsblockinstanz mit Werten zu belegen, geben Sie die einzelnen Elemente bzw. Instanzvariablen und deren Werte explizit ein.

Beispiel

Sie haben eine Struktur `STRU` mit den Komponenten a, b, c definiert, und eine Strukturvariable `struvar` im Programm `PLC_PRG` deklariert. Um die Komponenten a, b und c mit Werten vorzubelegen, geben Sie diese in der Watchliste wie folgt ein:

```
PLC_PRG.struvar.a:=<Wert>
PLC_PRG.struvar.b:=<Wert>
PLC_PRG.struvar.c:=<Wert>
```

Ebenso belegen Sie die Komponenten eines Arrays wie folgt vor:

Beispiel

Array-Variable `arr_var` vom Typ `ARRAY[0...6]`:

```
PLC_PRG.arr_var[0]:=<Wert>
PLC_PRG.arr_var[1]:=<Wert>
...
```

Wenn ein Funktionsblock `fb` die Variablen `x` und `y` enthält und die Instanzvariable `fb_inst` vom Typ `fb` im Programm `PLC_PRG` deklariert ist, belegen Sie die Variablen `x` und `y` wie folgt vor:

```
PLC_PRG.fb_inst.x:=<Wert>
PLC_PRG.fb_inst.y:=<Wert>
```

6.8.3 [Einfügen] / [Neue Watchliste]

Im Offline-Modus fügen Sie im Watch- und Rezepturverwalter mit dem Menübefehl [Einfügen] / [Neue Watchliste] ([Insert] / [New Watch List]) eine neue Watchliste ein. Geben Sie im erscheinenden Dialog den gewünschten Namen der Watchliste ein.

6.8.4 [Extras] / [Watchliste umbenennen]

Mit dem Menübefehl [Extras] / [Watchliste umbenennen] ([Extras] / [Rename Watch List]) ändern Sie den Namen einer Watchliste im Watch- und Rezepturverwalter.

Geben Sie im erscheinenden Dialog den neuen Namen der Watchliste ein.

6.8.5 [Extras] / [Watchliste speichern]

Mit dem Menübefehl [Extras] / [Watchliste speichern] ([Extras] / [Save Watch List]) speichen Sie die aktuelle Watchliste. Der Dialog zum Speichern einer Datei wird geöffnet. Der Dateiname ist vorbelegt mit dem Namen der Watchliste und erhält die Erweiterung `.wtc`.

Die gespeicherte Watchliste können Sie mit dem Befehl [Extras] / [Watchliste laden] ([Extras] / [Load Watch List]) wieder laden.

6.8.6 [Extras] / [Watchliste laden]

Mit dem Menübefehl [Extras] / [Watchliste laden] ([Extras] / [Load Watch List]) laden Sie eine gespeicherte Watchliste wieder. Der Dialog zum Öffnen einer Datei wird geöffnet. Wählen Sie die gewünschte Datei mit der Erweiterung `.wtc`. Im erscheinenden Dialog können Sie der Watchliste einen neuen Namen geben. Vorbelegt ist der Dateiname ohne Erweiterung.

Mit dem Menübefehl [Extras] / [Watchliste speichern] ([Extras] / [Save Watch List]) speichern Sie eine Watchliste.

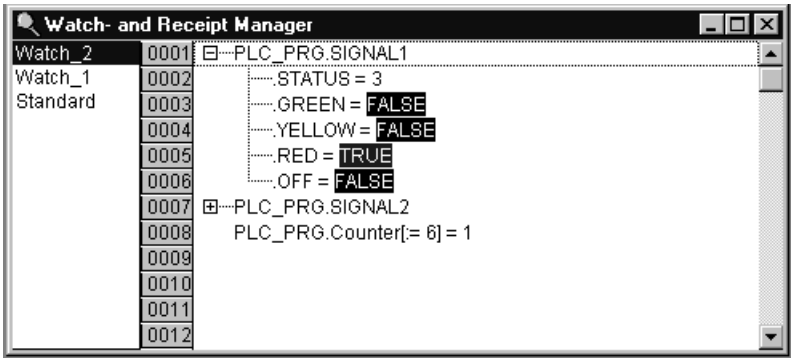
6.8.7 Watch- und Rezepturverwalter im Online-Modus

Im Online-Modus werden die Werte der eingegebenen Variablen angezeigt.

Strukturierte Werte wie Arrays, Strukturen oder Instanzen von Funktionsblöcken sind durch ein Pluszeichen vor dem Bezeichner gekennzeichnet. Mit einem Mausklick auf das Pluszeichen oder durch Drücken der <Eingabe>-Taste, klappen Sie die Variable auf und die einzelnen Komponenten werden angezeigt.

Wenn eine Funktionsblockvariable in der Watchliste ausgewählt ist, wird das zugehörige Kontextmenü um die beiden Menüpunkte [Zoom] ([Zoom]) und [Instanz öffnen] ([Open instance]) erweitert.

Um neue Variablen einzugeben, können Sie die Anzeige mit dem Befehl [Extra] / [Monitoring aktiv] ([Extra] / [Monitoring aktive]) ausschalten. Nach der Eingabe der Variablen, können Sie mit demselben Befehl das Anzeigen der Werte wieder aktivieren.



20542AEN

Im Offline-Modus können Sie Variablen mit konstanten Werten vorbelegen. Geben Sie dazu nach der Variablen das Zuweisungszeichen `:=` und den Wert ein.

Im Online-Modus schreiben Sie die Vorgabewerte mit dem Menübefehl [Extras] / [Rezeptur schreiben] ([Extras] / [Write recipe]) in die zugehörigen Variablen.

Weitere Informationen zu Array- und Strukturvariablen finden Sie auf Seite 322 (→ Kap. Watch- und Rezepturverwalter im Online-Modus).

Wenn eine Variable im Offline-Modus vorbelegt wurde, können Sie mit dem Menübefehl [Extras] / [Read recipe] deren Vorbelegung durch den aktuellen Wert der Variablen ersetzen.

Hinweis:

Es werden nur die Werte einer Watchliste geladen, die im Watch- und Rezepturverwalter ausgewählt wurden!

6.8.8 [Extras] / [Monitoring aktiv]

Mit dem Menübefehl [Extras] / [Monitoring aktiv] ([Extras] / [Monitoring aktive]) schalten Sie die Anzeige des Watch- und Rezepturverwalter im Online-Modus ein- bzw. aus. Wenn die Anzeige eingeschaltet ist, erscheint vor dem Menüpunkt ein Haken.

Um neue Variablen einzugeben oder einen Wert vorzubelegen, müssen Sie die Anzeige ausschalten. Nach der Eingabe der Variablen, können Sie das Anzeigen der Werte mit demselben Befehl wieder aktivieren.

6.8.9 [Extras] / [Rezeptur schreiben]

Mit dem Menübefehl [Extras] / [Rezeptur schreiben] ([Extras] / [Write recipe]) schreiben Sie im Online-Modus des Watch- und Rezepturverwalters die vorbelegten Werte in die zugehörigen Variablen.

Informationen dazu finden Sie auf Seite 178 (→ Kap. 4.6, Allgemeine Online-Funktionen).

Hinweis:

Es werden nur die Werte einer Watchliste geladen, die im Watch- und Rezepturverwalter ausgewählt wurden!

6.8.10 [Extras] / [Rezeptur lesen]

Mit dem Menübefehl [Extras] / [Rezeptur lesen] ([Extras] / [Read recipe]) ersetzen Sie im Online-Modus des Watch- und Rezepturverwalters die Vorbelegung der Variablen durch den aktuellen Wert der Variablen.

Informationen dazu finden Sie auf Seite 178 (→ Kap. 4.6, Allgemeine Online-Funktionen).

Beispiel

```
PLC_PRG.Zaehler [:= <aktueller Wert>] = <aktueller Wert>
```

Hinweis:

Es werden nur die Werte einer Watchliste geladen, die im Watch- und Rezepturverwalter ausgewählt wurden!

6.8.11 Werte forcen und schreiben im Watch Manager

Im Watch- und Rezepturverwalter können Sie mit den Menübefehlen [Werte forcen] ([Force value]) und [Werte schreiben] ([Write value]) Werte "forcen" bzw. Werte schreiben. Wenn Sie auf den jeweiligen Variablenwert klicken, erscheint ein Dialog, in dem Sie den neuen Wert der Variablen eingeben.

6.9 Zielsystem Einstellungen

6.9.1 Übersicht

Die Zielsystemeinstellungen finden Sie in der Registerkarte [Ressourcen] ([Resources]) des **Object Organizer**. Hier legen Sie fest, auf welcher Steuerung (Zielsystem, Target) und mit welchen Einstellungen das Projekt laufen soll. Wenn Sie den Befehl [Datei] / [Neu] ([File] / [New]) ausführen, werden Sie zur Auswahl einer vordefinierten Konfiguration eines Zielsystems aufgefordert.

Die Auswahlliste hängt von den installierten Target Support Packages (TSP) ab. Diese beschreiben plattformspezifische Grundkonfigurationen und legen gleichzeitig fest, wie weit der Anwender in den Dialogen der Zielsystemeinstellungen die Grundkonfiguration noch anpassen kann.

Die Target Support Packages werden automatisch während der Installation der Software MOVITOOLS® MotionStudio installiert.

Weitere Informationen finden Sie auf Seite 325 (→ Kap. 6.9.2, Dialog [Zielsystemeinstellungen]).

6.9.2 Dialog [Zielsystemeinstellungen]

Der Dialog [Zielsystemeinstellungen] ([Target Settings]) erscheint automatisch, wenn Sie ein neues Projekt anlegen. Diesen Dialog können Sie auch öffnen, indem Sie auf [Zielsystemeinstellungen] ([Target Settings]) in der Registerkarte [Ressourcen] ([Resources]) des Object Organizer doppelklicken.

Wählen Sie im Feld [Konfiguration] ([Configuration]) eine der verfügbaren Zielsystemkonfigurationen aus.

Wenn kein **Target Support Package** installiert ist, steht nur die Einstellung [None] zur Auswahl, die automatisch in den Simulationsmodus führt.

Wenn Sie eine der installierten Vorkonfigurationen auswählen, hängen die zur Verfügung stehenden Anpassungsmöglichkeiten von den Einträgen in der zugrundeliegenden Target-Datei ab. Wenn Sie eine Zielsystemkonfiguration aus einem Target Support Package auswählen, für das keine gültige Lizenz auf dem Rechner vorliegt, werden Sie zur Auswahl eines anderen Targets aufgefordert.

Wenn Sie eine Konfiguration einstellen, die in der Target-Datei mit "HideSettings" gekennzeichnet ist, erscheint nur der Name der Konfiguration. Ansonsten stehen die folgenden fünf Registerkarten zur Anpassung bzw. Darstellung der Zielsystemeinstellungen zur Verfügung:

- [Zielplattform] ([Target Platform]), Seite 326
- [Speicheraufteilung] ([Memory Layout]), Seite 327
- [Allgemein] ([General]), Seite 328
- [Netzfunktionen] ([Network funktionality]), Seite 329
- [Visualisierung] ([Visualization]), Seite 330

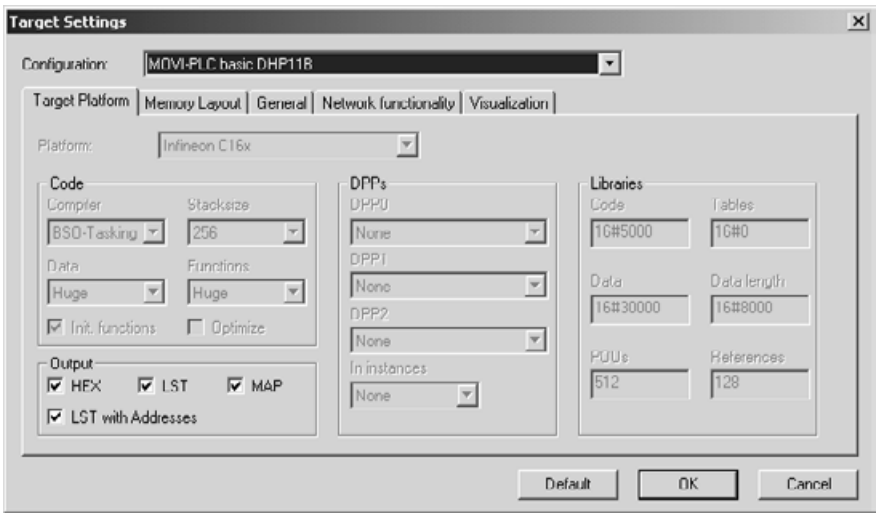
Achtung:

Beachten Sie, dass jede Veränderung der voreingestellten Zielsystemkonfiguration gravierende Auswirkungen auf das Verhalten des Zielsystems haben kann!

Wenn Sie auf die Schaltfläche [Voreinstellung] ([Default]) klicken, setzen Sie die Einstellungen nach einer Veränderung wieder auf die Werte der Standardkonfiguration zurück.

6.9.3 Zielsystemeinstellungen DHP11B

Zielsystemeinstellungen, Zielplattform



20543AEN

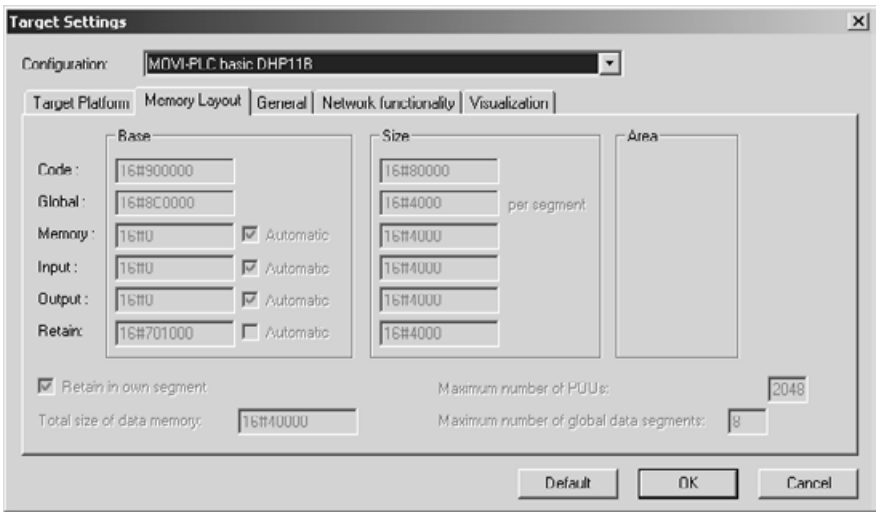
In der Registerkarte [Zielplattform] ([Target Platform]) des Dialogs [Zielsystemeinstellungen] ([Target Settings]) können Sie folgende Einstellungen vornehmen:

[Output]	Hier aktivieren Sie die Ausgabe des Codes in verschiedenen Dateiformaten. • [HEX]: Der Code wird in einer HEX-Datei ausgegeben. • [MAP]: Der Code wird in einer MAP-Datei ausgegeben. • [LST]: Der Code wird in einer List-Datei ausgegeben. • [LST mit Adressen] ([LST with Addresses]): Die Code-Adressen werden in einer List-Datei ausgegeben.

Weitere Informationen finden Sie in den folgenden Kapiteln bzw. Abschnitten:

- (→ Kap. 6.9, Zielsystem Einstellungen), Seite 325
- (→ Abschnitt. Zielsystemeinstellungen, Speicheraufteilung), Seite 327
- (→ Abschnitt. Zielsystemeinstellungen, Allgemein), Seite 328
- (→ Abschnitt. Zielsystemeinstellungen, Netzfunktion), Seite 329
- (→ Abschnitt. Zielsystemeinstellungen, Visualisierung), Seite 330

Zielsystemeinstellungen, Speicheraufteilung



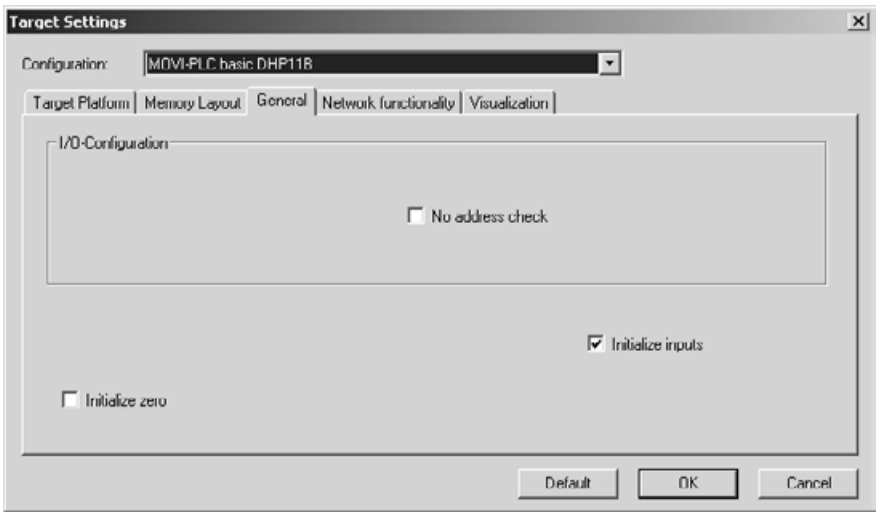
20544AEN

Die Einstellungen in der Registerkarte [Speicheraufteilung] ([Memory Layout]) des Dialogs [Zielsystemeinstellungen] ([Target Settings]) sind in der Konfigurationsdatei festgelegt und können nicht verändert werden.

Weitere Informationen finden Sie in den folgenden Kapiteln bzw. Abschnitten:

- (→ Kap. 6.9, Zielsystem Einstellungen), Seite 325
- (→ Abschnitt. Zielsystemeinstellungen, Zielplattform), Seite 326
- (→ Abschnitt. Zielsystemeinstellungen, Allgemein), Seite 328
- (→ Abschnitt. Zielsystemeinstellungen, Netzfunktion), Seite 329
- (→ Abschnitt. Zielsystemeinstellungen, Visualisierung), Seite 330

Zielsystemeinstellungen, Allgemein



20545AEN

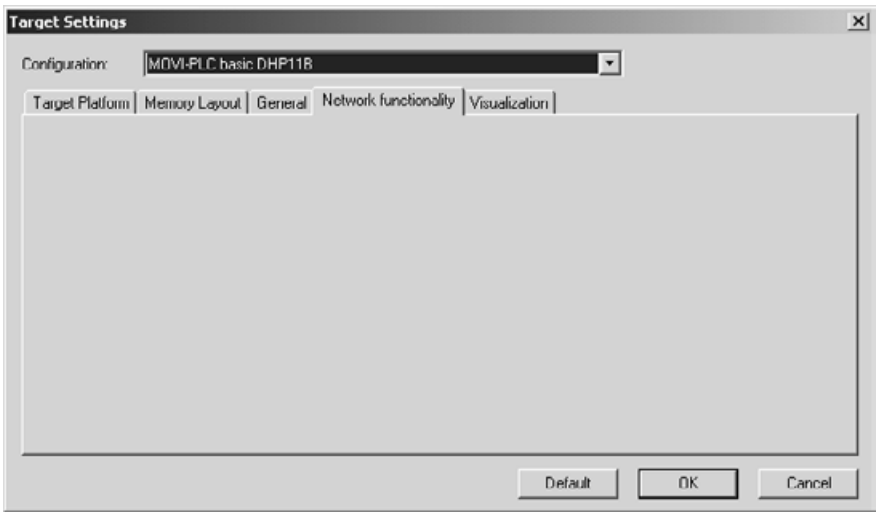
In der Registerkarte [Allgemein] ([General]) des Dialogs [Zielsystemeinstellungen] ([Target Settings]) können Sie folgende Einstellungen vornehmen:

[Keine Adressprüfung]	[No address check]	Wenn diese Option aktiviert ist, werden die IEC-Adressen des Projekts beim Übersetzen nicht geprüft.
[Nullinitialisierung]	[Intialize zero]	Wenn diese Option aktiviert ist, erfolgt eine generelle Initialisierung mit Null.
[Eingänge initialisieren]	[Initialize inputs]	Wenn diese Option nicht aktiviert ist, wird aus Optimierungsgründen kein Initialisierungscode für die mit `AT %IX` deklarierten Eingänge erzeugt. Die Werte sind bis zum ersten Buszyklus undefiniert!

Weitere Informationen finden Sie in den folgenden Kapiteln bzw. Abschnitten:

- (→ Kap. 6.9, Zielsystem Einstellungen), Seite 325
- (→ Abschnitt. Zielsystemeinstellungen, Zielplattform), Seite 326
- (→ Abschnitt. Zielsystemeinstellungen, Speicheraufteilung), Seite 327
- (→ Abschnitt. Zielsystemeinstellungen, Netzfunktion), Seite 329
- (→ Abschnitt. Zielsystemeinstellungen, Visualisierung), Seite 330

Zielsystemeinstellungen, Netzfunktion



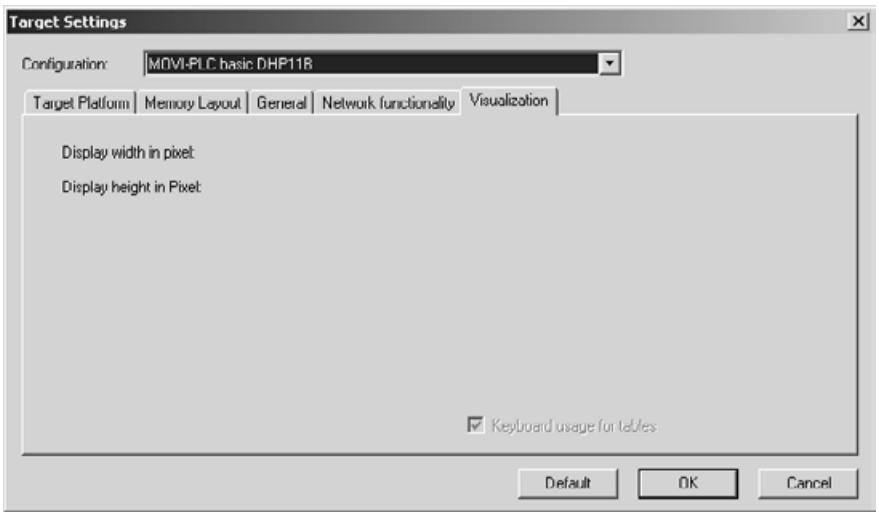
20546AEN

Die Einstellungen in der Registerkarte [Netzfunktionen] ([Network functionality]) des Dialogs [Zielsystemeinstellungen] ([Target Settings]) können nicht verändert werden.

Weitere Informationen finden Sie den in folgenden Kapiteln bzw Abschnitten:

- (→ Kap. 6.9, Zielsystem Einstellungen), Seite 325
- (→ Abschnitt. Zielsystemeinstellungen, Zielplattform), Seite 326
- (→ Abschnitt. Zielsystemeinstellungen, Speicheraufteilung), Seite 327
- (→ Abschnitt. Zielsystemeinstellungen, Allgemein), Seite 328
- (→ Abschnitt. Zielsystemeinstellungen, Visualisierung), Seite 330

Zielsystemeinstellungen, Visualisierung



20547AEN

Die Einstellungen in der Registerkarte [Visualisierung] ([Visualization]) des Dialogs [Zielsystemeinstellungen] ([Target Settings]) sind in der Konfigurationsdatei festgelegt und können nicht verändert werden.

Weitere Informationen finden Sie in den folgenden Kapiteln bzw. Abschnitten:

- (→ Kap. 6.9, Zielsystem Einstellungen), Seite 325
- (→ Abschnitt. Zielsystemeinstellungen, Zielplattform), Seite 326
- (→ Abschnitt. Zielsystemeinstellungen, Speicheraufteilung), Seite 327
- (→ Abschnitt. Zielsystemeinstellungen, Allgemein), Seite 328
- (→ Abschnitt. Zielsystemeinstellungen, Netzfunktion), Seite 329

7 Operatoren und Funktionen

7.1 Überblick

Der PLC-Editor unterstützt alle IEC-Operatoren. Diese sind, im Gegensatz zu den Standardfunktionen in der Standardbibliothek, implizit im ganzen Projekt bekannt. Neben den IEC-Operatoren unterstützt der PLC-Editor außerdem folgende nicht von der Norm geforderten Operatoren:

- `INDEXOF` und `SIZEOF`, Seite 332 (→ Kap. 7.2, Arithmetische Operatoren).
- `ADR` und `BITADR`, Seite 357 (→ Kap. 7.7, Adressoperatoren)

Hinweis:

Bei Operationen mit Gleitkomma-Datentypen ist das Rechenergebnis von der verwendeten Zielsystem-Hardware abhängig!

In den Baustein-Implementationen werden Operatoren wie Funktionen benutzt.

- (→ Kap. 7.2, Arithmetische Operatoren), Seite 332
- (→ Kap. 7.3, Bitstring Operatoren), Seite 339
- (→ Kap. 7.4, Bit-Shift Operatoren), Seite 342
- (→ Kap. 7.5, Auswahloperatoren), Seite 346
- (→ Kap. 7.6, Vergleichsoperatoren), Seite 351
- (→ Kap. 7.7, Adressoperatoren), Seite 357
- (→ Kap. 7.8, Aufrufoperator), Seite 359
- (→ Kap. 7.9, Typkonvertierungen), Seite 360
- (→ Kap. 7.10, Nummerische Operatoren), Seite 370

7.2 Arithmetische Operatoren

7.2.1 ADD

Operator: ADD oder +

IEC-Operator: Ja

Verwendung: Addition von Variablen

Variablentypen:

- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT
- REAL, LREAL
- TIME

Wenn zwei TIME-Variablen addiert werden, ist die Summe dann wieder eine TIME-Variable (z. B. gilt: `t#45s + t#50s = t#1m35s`).

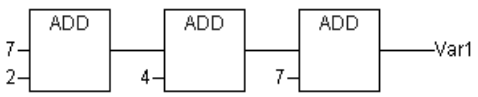
Beispiel in AWL

```
LD  7
ADD 2,4,7
ST  Var1
```

Beispiel in ST

```
Var1 := 7+2+4+7;
```

Beispiel in FUP



20602AXX

7.2.2 MUL

Operator: MUL oder *

IEC-Operator: Ja

Verwendung: Multiplikation von Variablen

Variablentypen:

- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT
- REAL, LREAL

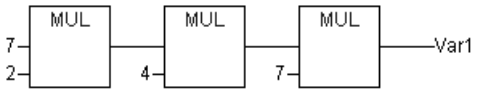
Beispiel in AWL

```
LD  7
MUL 2,4,7
ST  Var1
```

Beispiel in ST

```
Var1 := 7*2*4*7;
```

Beispiel in FUP



20603AXX

7.2.3 SUB

Operator: SUB oder -

IEC-Operator: Ja

Verwendung: Subtraktion von zwei Variablen

Variablentypen:

- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT
- REAL, LREAL
- TIME

Wenn eine TIME-Variable von einer anderen TIME-Variablen subtrahiert wird, ist das Ergebnis wieder eine TIME-Variable.

Hinweis:

Negative TIME-Werte sind nicht definiert!

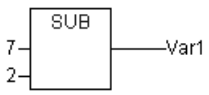
Beispiel in AWL

```
LD  7
SUB 2
ST  Var1
```

Beispiel in ST

```
Var1 := 7-2
```

Beispiel in FUP



20604AXX

7.2.4 DIV

Operator: DIV oder /

IEC-Operator: Ja

Verwendung: Division von zwei Variablen

Variablentypen:

- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT
- REAL, LREAL

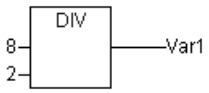
Beispiel in AWL

```
LD  8
DIV 2
ST  Var1 (* Ergebnis ist 4 *)
```

Beispiel in ST

```
Var1 := 8/2;
```

Beispiel in FUP



20605AXX

Hinweis:

Wenn Sie in Ihrem Projekt Funktionen mit Namen CheckDivByte, CheckDivWord, CheckDivDWord und CheckDivReal definieren, können Sie damit bei Verwendung des Operators DIV den Wert des Divisors überprüfen. Damit können Sie beispielsweise eine Division durch 0 verhindern.

Die Namen der Funktionen sind festgelegt und dürfen nur diese Bezeichnungen besitzen.

Hinweis:

Das Verhalten im Falle einer Division durch 0 ist vom eingesetzten Betriebs- und Zielsystem abhängig.

Beispiel für die Implementierung der Funktion CheckDivReal

```
FUNCTION CheckDivReal : REAL
   VAR_INPUT
      divisor : REAL;
   END_VAR
   IF divisor = 0 THEN
      CheckDivReal := 1;
   ELSE
      CheckDivReal := divisor;
   END_IF;
```

Das Ergebnis der Funktion `CheckDivReal` wird vom Operator `DIV` als Divisor eingesetzt. Im folgenden Beispielprogramm wird dadurch eine Division durch 0 verhindert. Der Divisor d wird von 0 auf 1 gesetzt. Das Ergebnis `erg` der Division ist somit 799.

```
PROGRAM PLC_PRG
   VAR
      erg : REAL;
      v1 : REAL := 799;
      d : REAL;
   END_VAR
   erg := v1/d
```

Hinweis:

Die in der Bibliothek `check.lib` enthaltenen CheckDiv-Funktionen sind Beispiellösungen!

Prüfen Sie vor Verwendung der Bibliothek `check.lib`, ob die Funktionen wie gewünscht arbeiten oder implementieren Sie eine entsprechende CheckDiv-Funktion direkt als Baustein in Ihrem Projekt.

7.2.5 MOD

Operator: `MOD`

IEC-Operator: Ja

Verwendung: Modulo-Division von zwei Variablen. Das Ergebnis ist der ganzzahlige Rest der Division.

Variablentypen:

- `BYTE`
- `WORD`, `DWORD`
- `INT`, `UINT`, `DINT`, `UDINT`, `SINT`, `USINT`

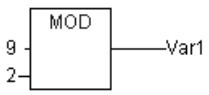
Beispiel in AWL

```
LD   9
MOD  2
ST   Var1 (* Ergebnis ist 1 *)
```

Beispiel in ST

```
Var1 := 9 MOD 2;
```

Beispiel in FUP



20606AXX

7.2.6 MOVE

Operator: MOVE

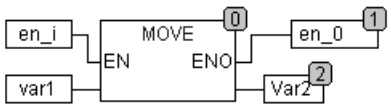
IEC-Operator: Ja

Verwendung: Zuweisung einer Variablen auf eine andere Variable eines entsprechenden Typs.

Da der Operator MOVE in den CFC- und KOP-Editoren als Baustein verfügbar ist, kann die EN/ENO-Funktionalität dort auch auf eine Variablenzuweisung angewendet werden. Im FUP-Editor ist dies nicht möglich.

Beispiel in CFC in Verbindung mit der EN/EN0 Funktion

Nur wenn der Eingang en_i auf TRUE gesetzt ist, wird der Wert der Variablen var1 der Variablen var2 zugewiesen.



20607AXX

Beispiel in AWL

```
LD ivar1
MOVE
ST ivar2  (* Ergebnis: var2 erhält den Wert von var1 *)
```

Das Beispiel ohne die Verwendung von MOVE:

```
LD ivar1
ST ivar2
```

Beispiel in ST

```
ivar2 := MOVE(ivar1);
```

Das Beispiel ohne die Verwendung von MOVE:

```
ivar2 := ivar1;
```

7.2.7 INDEXOF

Operator: INDEXOF

IEC-Operator: Diese Funktion wird nicht von der Norm IEC 61131-3 gefordert.

Verwendung: Der Operator INDEXOF liefert als Ergebnis den internen Index eines Bausteins.

Beispiel in ST

```
Var1 := INDEXOF(baustein2);
```

7.2.8 SIZEOF

Operator: SIZEOF

IEC-Operator: Diese Funktion wird nicht von der Norm IEC 61131-3 gefordert.

Verwendung: Der Operator SIZEOF liefert als Ergebnis die Anzahl der Bytes, die die Variable benötigt..

Beispiel in AWL

```
arr1 : ARRAY[0..4] OF INT;
Var1 INT
LD arr1
SIZEOF
ST Var1 (* Ergebnis ist 10 *)
```

Beispiel in ST

```
Var1 := SIZEOF(arr1);
```

7.3 Bitstring Operatoren

7.3.1 AND

Operator: AND

IEC-Operator: Ja

Verwendung: Bitweises UND von Bit-Operanden.

Wenn zwei Bits den Wert "1" haben, liefert der Operator AND das Ergebnis "1", in allen anderen Fällen liefert der Operator das Ergebnis "0".

Variablentypen:

- BOOL
- BYTE
- WORD, DWORD

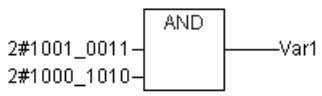
Beispiel in AWL

```
Var1 BYTE
LD   2#1001_0011
AND  2#1000_1010
ST   Var1 (* Ergebnis ist 2#1000_0010 *)
```

Beispiel in ST

```
Var1 := 2#1001_0011 AND 2#1000_1010
```

Beispiel in FUP



20608AXX

7.3.2 OR

Operator: OR

IEC-Operator: Ja

Verwendung: Bitweises ODER von Bit-Operanden.

Wenn zwei Bits den Wert "0" haben, liefert der Operator OR das Ergebnis "0", in allen anderen Fällen liefert der Operator das Ergebnis "1".

Variablentypen:

- BOOL
- BYTE
- WORD, DWORD

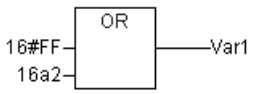
Beispiel in AWL

```
Var1 BYTE
LD   2#1001_0011
OR   2#1000_1010
ST   Var1    (* Ergebnis ist 2#1001_1011 *)
```

Beispiel in ST

```
Var1 := 2#1001_0011 OR 2#1000_1010
```

Beispiel in FUP



20610AXX

7.3.3 XOR

Operator: XOR

IEC-Operator: Ja

Verwendung: Bitweises EXLUSIVES ODER von Bit-Operanden.

Wenn zwei Bits unterschiedliche Werte haben, liefert der Operator XOR das Ergebnis "1", wenn zwei Bits den gleichen Wert haben, liefert der Operator den Wert "0".

Variablentypen:

- BOOL
- BYTE
- WORD, DWORD

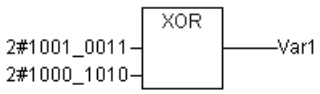
Beispiel in AWL

```
Var1 BYTE
LD   2#1001_0011
XOR  2#1000_1010
ST   Var1    (* Ergebnis ist 2#0001_1001 *)
```

Beispiel in ST

```
Var1 := 2#1001_0011 XOR 2#1000_1010
```

Beispiel in FUP



20612AXX

Hinweis:

Beachten Sie das Verhalten des Baustein XOR, wenn der Baustein mehr als zwei Eingänge hat!

Die Eingänge werden paarweise geprüft und die jeweiligen Ergebnisse wiederum miteinander verglichen. Dies entspricht der Norm, jedoch nicht unbedingt der Erwartung.

7.3.4 NOT

Operator: NOT

IEC-Operator: Ja

Verwendung: Bitweise NEGATION von Bit-Operanden.

Wenn ein Bit den Wert "0" haben, liefert der Operator OR das Ergebnis "1" und umgekehrt.

Variablentypen:

- BOOL
- BYTE
- WORD, DWORD

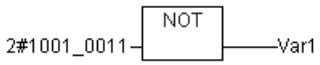
Beispiel in AWL

```
Var1 BYTE
LD   2#1001_0011
NOT
ST   Var1    (* Ergebnis ist 2#0110_1100 *)
```

Beispiel in ST

```
Var1 := NOT 2#1001_0011
```

Beispiel in FUP



20613AXX

7.4 Bit-Shift Operatoren

7.4.1 SHL

Operator: SHL (shift left)

IEC-Operator: Ja

Verwendung: Bitweises Verschieben eines Operanden nach links.

Syntax: `erg:= SHL (in, n)`

Der Eingang `in` wird um n Bits nach links geschoben und von rechts mit Nullen aufgefüllt.

Hinweis:

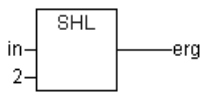
Beachten Sie, dass die Anzahl der Bits, die für die Rechenoperation berücksichtigt wird, durch den Datentyp der Eingangsvariablen in vorgegeben wird. Wenn es sich hierbei um eine Konstante handelt, wird der kleinstmögliche Datentyp berücksichtigt. Der Datentyp der Ausgangsvariable bleibt ohne Auswirkung auf die Rechenoperation.

Beachten Sie im folgenden Beispiel in hexadezimaler Darstellung, wie sich bei gleichem Wert der Eingangsvariablen `in_byte` und `in_word` die Ergebnisse `erg_byte` und `erg_word` der Operation unterscheiden, je nachdem, ob die beiden Eingangsvariablen vom Typ BYTE oder WORD sind.

Beispiel in ST

```
PROGRAM shl_st
   VAR
      in_byte : BYTE:=16#45;
      in_word : WORD:=16#45;
      erg_byte : BYTE;
      erg_word : WORD;
      n: BYTE := 2;
   END_VAR
   erg_byte := SHL(in_byte,n); (* Ergebnis ist 16#14 *)
   erg_word := SHL(in_word;n); (* Ergebnis ist 16#0114 *)
```

Beispiel in FUP



20614AXX

Beispiel in AWL

```
LD  16#45
SHL 2
ST  erg_byte
```

7.4.2 SHR

Operator: SHR (shift right)

IEC-Operator: Ja

Verwendung: Bitweises Verschieben eines Operanden nach rechts.

Syntax: `erg:= SHR (in, n)`

Der Eingang `in` wird um `n` Bits nach rechts geschoben.

Bit-Shift IEC-Operator: Bitweise Rechts-Shift eines Operanden: erg:= SHR (*in*, *n*)

Bei Datentyp ohne Vorzeichen (z. B. `BYTE`, `WORD`, `DWORD`) wird von links mit Nullen aufgefüllt.

Bei Datentypen mit Vorzeichen (z. B. `INT`) wird ein arithmetischer Shift durchgeführt, von links wird mit dem Wert des obersten Bits aufgefüllt.

Hinweis:

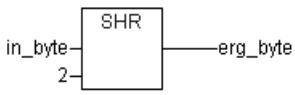
Beachten Sie, dass die Anzahl der Bits, die für die Rechenoperation berücksichtigt wird, durch den Datentyp der Eingangsvariablen `in` vorgegeben wird. Wenn es sich hierbei um eine Konstante handelt, wird der kleinstmögliche Datentyp berücksichtigt. Der Datentyp der Ausgangsvariable bleibt ohne Auswirkung auf die Rechenoperation.

Beachten Sie im folgenden Beispiel in hexadezimaler Darstellung die Ergebnisse der Operation, wobei einmal das Ergebnis `erg_byte` vom Typ `BYTE` als Eingangsvariable dient und einmal das Ergebnis `erg_word` vom Typ `WORD` .

Beispiel in ST

```
PROGRAM shr_st
  VAR
     in_byte : BYTE:=16#45;
     in_word : WORD:=16#45;
     erg_byte : BYTE;
     erg_word : WORD;
     n : BYTE := 2;
  END_VAR
  erg_byte := SHR(in_byte,n); (* Ergebnis ist 11 *)
  erg_word := SHR(in_word;n); (* Ergebnis ist 0011 *)
```

Beispiel in FUP



20615AXX

Beispiel in AWL

```
LD  16#45
SHR 2
ST  erg_byte
```

7.4.3 ROL

Operator: ROL (rotation left)

IEC-Operator: Ja

Verwendung: Bitweise Linksrotation eines Operanden

Syntax: `erg := ROL (in, n)`

Der Einhang `in` wird `n` mal um eine Bitstelle nach links geschoben. Das Bit ganz links wird von rechts wieder eingeschoben.

Variablentypen (`erg`, `in` und `n`):

- `BYTE`
- `WORD`, `DWORD`

Hinweis:

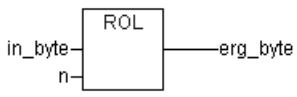
Beachten Sie, dass die Anzahl der Bits, die für die Rechenoperation berücksichtigt wird, durch den Datentyp der Eingangsvariable `in` vorgegeben wird. Wenn es sich hierbei um eine Konstante handelt, wird der kleinstmögliche Datentyp berücksichtigt. Der Datentyp der Ausgangsvariable bleibt ohne Auswirkung auf die Rechenoperation.

Beachten Sie im folgenden Beispiel in hexadezimaler Darstellung, dass sich bei gleichem Wert der Eingangsvariablen `in_byte` und `in_word` die Ergebnisse `erg_byte` und `erg_word` der Operation unterscheiden, je nachdem, ob sie vom Typ `BYTE` oder `WORD` sind.

Beispiel in ST

```
PROGRAM rol_st
  VAR
     in_byte : BYTE:=16#45;
     in_word : WORD:=16#45;
     erg_byte : BYTE;
     erg_word : WORD;
     n : BYTE := 2;
  END_VAR
  erg_byte := ROL(in_byte,n); (* Ergebnis ist 16#15 *)
  erg_word := ROL(in_word,n); (* Ergebnis ist 16#0114 *)
```

Beispiel in FUP



20616AXX

Beispiel in AWL

```
LD  16#45
ROL 2
ST  erg_byte
```

7.4.4 ROR

Operator: ROR (rotation right)

IEC-Operator: Ja

Verwendung: Bitweise Rechtsrotation eines Operanden

Syntax: `erg := ROR (in, n)`

Der Einhang `in` wird `n` mal um eine Bitstelle nach rechts geschoben. Das Bit ganz rechts wird von links wieder eingeschoben.

Variablentypen (`erg`, `in` und `n`):

- `BYTE`
- `WORD`, `DWORD`

Hinweis:

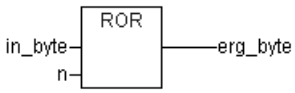
Beachten Sie, dass die Anzahl der Bits, die für die Rechenoperation berücksichtigt wird, durch den Datentyp der Eingangsvariable `in` vorgegeben wird. Wenn es sich hierbei um eine Konstante handelt, wird der kleinstmögliche Datentyp berücksichtigt. Der Datentyp der Ausgangsvariable bleibt ohne Auswirkung auf die Rechenoperation.

Beachten Sie im folgenden Beispiel in hexadezimaler Darstellung, dass sich bei gleichem Wert der Eingangsvariablen `in_byte` und `in_word` die Ergebnisse `erg_byte` und `erg_word` der Operation unterscheiden, je nachdem, ob sie vom Typ `BYTE` oder `WORD` sind.

Beispiel in ST

```
PROGRAM ror_st
  VAR
     in_byte : BYTE:=16#45;
     in_word : WORD:=16#45;
     erg_byte : BYTE;
     erg_word : WORD;
     n : BYTE := 2;
  END_VAR
  erg_byte := ROR(in_byte,n); (* Ergebnis ist 16#51 *)
  erg_word := ROR(in_word,n); (* Ergebnis ist 16#4011 *)
```

Beispiel in FUP



20617AXX

Beispiel in AWL

```
LD    16#45
ROR   2
ST    erg_byte
```

7.5 Auswahloperatoren

7.5.1 Hinweise

Alle Auswahloperationen lassen sich auch auf Variablen durchführen. Wegen der besseren Anschaulichkeit werden im Folgenden nur Beispiele gezeigt, in denen Konstanten als Operatoren verwendet werden.

7.5.2 SEL

Operator: SEL (selection)

IEC-Operator: Ja

Verwendung: Binäre Selektion

Syntax: `OUT := SEL(G, IN0, IN1)`

Der Operator SEL liefert einen der beiden Eingänge.

Das Beispiel ohne die Verwendung von SEL:

```
OUT := IN0 if G=FALSE;
OUT := IN1 if G=TRUE;
```

Variablentypen: Die Variablen IN0, IN1 und OUT können von beliebigem Typ sein. Der Eingang G muss vom Typ BOOL sein.

Das Ergebnis der Selektion ist IN0, wenn der Eingang G auf FALSE gesetzt ist, bzw. IN1, wenn der Eingang G auf TRUE gesetzt ist.

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

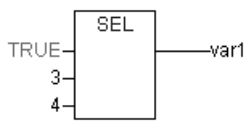
Beispiel in AWL

```
LD  TRUE
SEL 3,4   (* IN0 = 3, IN1 =4 *)
ST  Var1  (* Ergebnis ist 4 *)
LD  FALSE
SEL 3,4
ST  Var1  (* Ergebnis ist 3 *)
```

Beispiel in ST

```
Var1 := SEL(TRUE,3,4); (* Ergebnis für Var1 ist 4 *)
```

Beispiel in FUP



20618AXX

Hinweis:

Um die Laufzeit zu optimieren, wird der Operator wie folgt abgearbeitet: Ein Ausdruck, der IN0 vorgeschaltet ist, wird nur dann berechnet, wenn G auf FALSE gesetzt ist. Ein Ausdruck der dem Eingang IN1 vorgeschaltet ist, wird nur dann berechnet, wenn der Eingang G auf TRUE gesetzt ist.

Im Simulationsmodus werden dagegen alle Zweige berechnet.

7.5.3 MAX

Operator: MAX (maximum)

IEC-Operator: Ja

Verwendung: Maximumsfunktion

Syntax: `OUT := MAX(IN0, IN1)`

Der Operator MAX liefert den größeren der beiden Eingänge.

Variablentypen: Die Variablen IN0, IN1 und OUT können von beliebigem Typ sein. .

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

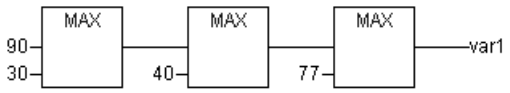
Beispiel in AWL

```
LD   90
MAX  30
MAX  40
MAX  77
ST   Var1 (* Ergebnis ist 90 *)
```

Beispiel in ST

```
Var1 := MAX(30,40); (* Ergebnis ist 40 *)
Var1 := MAX(40,MAX(90,30)); (* Ergebnis ist 90 *)
```

Beispiel in FUP



20619AXX

7.5.4 MIN

Operator: `MIN` (minimum)

IEC-Operator: Ja

Verwendung: Minimumsfunktion

Syntax: `OUT := MIN(IN0, IN1)`

Der Operator `MIN` liefert den kleineren der beiden Eingänge.

Variablentypen: Die Variablen `IN0`, `IN1` und `OUT` können von beliebigem Typ sein. .

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

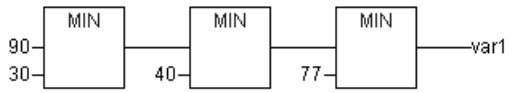
Beispiel in AWL

```
LD   90
MIN  30
MIN  40
MIN  77
ST   Var1 (* Ergebnis ist 30 *)
```

Beispiel in ST

```
Var1 := MIN(90,30); (* Ergebnis ist 30 *);
Var1 := MIN(MIN(90,30),40); (* Ergebnis ist 30 *);
```

Beispiel in FUP



20620AXX

7.5.5 LIMIT

Operator: LIMIT (limitation)

IEC-Operator: Ja

Verwendung: Limitierung

Syntax: OUT := LIMIT (Min, IN, Max)

Der Eingang Max ist die obere, der Eingang Min die untere Schranke für das Ergebnis. Wenn der Wert IN die obere Grenze Max überschreitet, dann liefert der Operator LIMIT den Wert Max. Wenn der Wert IN die untere Grenze Min unterschreitet, dann ist das Ergebnis gleich dem Wert Min.

Das Beispiel ohne die Verwendung von LIMIT:

```
OUT := MIN (MAX (IN, Min), Max)
```

Variablentypen: Die Variablen IN und OUT können von beliebigem Typ sein. .

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

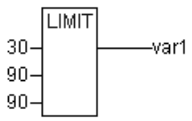
Beispiel in AWL

```
LD    90
LIMIT 30,80
ST    Var1 (* Ergebnis ist 80 *)
```

Beispiel in ST

```
Var1 := LIMIT(30,90,80); (* Ergebnis ist 80 *);
```

Beispiel in FUP



20621AXX

7.5.6 MUX

Operator: MUX (multiplexer)

IEC-Operator: Ja

Verwendung: Multiplexer

Syntax: `OUT := MUX (K, IN0, ... , INn)`

Der Operator MUX wählt aus einer Menge das K-te Element aus und weist dessen Wert der Variablen OUT zu. Das erste Element IN0 wird bei K=0 ausgewählt. Wenn K größer ist als die Anzahl der weiteren Eingänge n, wird der Wert von INn weitergegeben.

Das Beispiel ohne die Verwendung von MUX:

```
OUT := INk
```

Variablentypen:

Die Variablen IN0 ... INn und OUT können von beliebigem Typ sein.

Die Variable K muss einen der folgenden Typen haben:

- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

Beispiel in AWL

```
LD   0
MUX  30,40,50,60,70,80
ST   Var1 (* Ergebnis ist 30 *)
```

Beispiel in ST

```
Var1 := MUX(0,30,40,50,60,70,80); (* Ergebnis ist 30 *);
```

Hinweis:

Um die Laufzeit zu optimieren, wird nur der Ausdruck, der INk vorgeschaltet ist, berechnet!

Im Simulationsmodus werden dagegen alle Zweige berechnet.

7.6 Vergleichsoperatoren

7.6.1 GT

Operator: GT (greater than)

IEC-Operator: Ja

Verwendung: Der boolesche Operator GT (größer als) setzt den Ausgang auf TRUE, wenn der erste Operand größer ist als der zweite.

Variablentypen der Operanden:

- BOOL
- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT
- REAL, LREAL
- TIME, DATE, TIME_OF_DAY, DATE_AND_TIME
- STRING

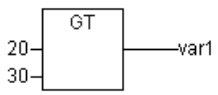
Beispiel in AWL

```
LD 20
GT 30
ST Var1 (* Ergebnis ist FALSE *)
```

Beispiel in ST

```
Var1 := 20 > 30 > 40 > 50 > 60 > 70;
```

Beispiel in FUP



20622AXX

7.6.2 LT

Operator: GT (lower than)

IEC-Operator: Ja

Verwendung: Der boolesche Operator LT (kleiner als) setzt den Ausgang auf TRUE, wenn der erste Operand kleiner ist als der zweite.

Variablentypen der Operanden:

- BOOL
- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT
- REAL, LREAL
- TIME, DATE, TIME_OF_DAY, DATE_AND_TIME
- STRING

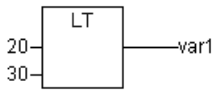
Beispiel in AWL

```
LD 20
LT 30
ST Var1 (* Ergebnis ist TRUE *)
```

Beispiel in ST

```
Var1 := 20 < 30;
```

Beispiel in FUP



20623AXX

7.6.3 GE

Operator: GE (greater or equal)

IEC-Operator: Ja

Verwendung: Der boolesche Operator GE (größer oder gleich) setzt den Ausgang auf TRUE, wenn der erste Operand größer als oder gleich groß wie der zweite Operand ist.

Variablentypen der Operanden:

- BOOL
- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT
- REAL, LREAL
- TIME, DATE, TIME_OF_DAY, DATE_AND_TIME
- STRING

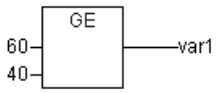
Beispiel in AWL

```
LD 60
GE 40
ST Var1 (* Ergebnis ist TRUE *)
```

Beispiel in ST

```
Var1 := 60 >= 40;
```

Beispiel in FUP



20625AXX

7.6.4 LE

Operator: LE (lower or equal)

IEC-Operator: Ja

Verwendung: Der boolesche Operator GT (kleiner oder gleich) setzt den Ausgang auf TRUE, wenn der erste Operand größer als oder gleich groß wie der zweite Operand ist.

Variablentypen der Operanden:

- BOOL
- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT
- REAL, LREAL
- TIME, DATE, TIME_OF_DAY, DATE_AND_TIME
- STRING

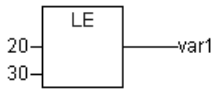
Beispiel in AWL

```
LD 20
LE 30
ST Var1 (* Ergebnis ist TRUE *)
```

Beispiel in ST

```
Var1 := 20 <= 30;
```

Beispiel in FUP



20624AXX

7.6.5 EQ

Operator: EQ (equal)

IEC-Operator: Ja

Verwendung: Der boolesche Operator EQ (gleich) setzt den Ausgang auf TRUE, wenn die Operanden gleich sind.

Variablentypen der Operanden:

- BOOL
- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT
- REAL, LREAL
- TIME, DATE, TIME_OF_DAY, DATE_AND_TIME
- STRING

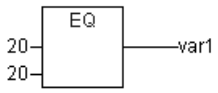
Beispiel in AWL

```
LD 40
EQ 40
ST Var1 (* Ergebnis ist TRUE *)
```

Beispiel in ST

```
Var1 := 40 = 40;
```

Beispiel in FUP



20626AXX

7.6.6 NE

Operator: NE (not equal)

IEC-Operator: Ja

Verwendung: Der boolesche Operator NE (nicht gleich) setzt den Ausgang auf TRUE, wenn die Operanden ungleich sind.

Variablentypen der Operanden:

- BOOL
- BYTE
- WORD, DWORD
- INT, UINT, DINT, UDINT, SINT, USINT
- REAL, LREAL
- TIME, DATE, TIME_OF_DAY, DATE_AND_TIME
- STRING

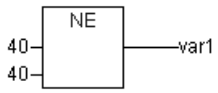
Beispiel in AWL

```
LD 40
NE 40
ST Var1 (* Ergebnis ist FALSE *)
```

Beispiel in ST

```
Var1 := 40 <> 40;
```

Beispiel in FUP



20627AXX

7.7 Adressoperatoren

7.7.1 ADR

Operator: ADR (address)

IEC-Operator: Nein, der Operator wird nicht von der Norm IEC 61131-3 gefordert.

Verwendung: Der Operator ADR liefert die Adresse seines Arguments als DWORD. Diese Adresse kann an die Herstellerfunktionen gesendet und dort wie ein Pointer behandelt werden. Außerdem kann die Adresse innerhalb des Projekts einem Pointer zugewiesen werden.

Beispiel in AWL

```
LD    bVar
ADR
ST    dwVar
man_fun1
```

Beispiel in ST

```
dwVar := ADR(bVAR);
```

Hinweis:

Wenn Sie **Online Change** anwenden, können sich Inhalte von Adressen verschieben. Beachten Sie dies bei der Verwendung von Pointern auf Adressen!

7.7.2 ADRINST

Operator: ADRINST (address instance)

IEC-Operator: Nein, der Operator wird nicht von der Norm IEC 61131-3 gefordert.

Verwendung: Der Operator ADRINST liefert innerhalb einer Funktionsblockinstanz die Adresse dieser Instanz als DWORD. Diese Adresse kann an Funktionen übergeben und dort wie ein Pointer behandelt werden. Außerdem kann sie innerhalb des Projekts einem Pointer zugewiesen werden.

Beispiele in AWL

```
ADRINST
ST dvar

ADRINST
fun
```

Beispiele in ST (innerhalb einer Funktionsblockinstanz)

```
dvar := ADRINST(); (* Adresse der Instanz auf Variable dvar schreiben *)
fun(a := ADRINST()); (* Übergabe der Instanzadresse
                        an Eingangsparameter a von Funktion fun *)
```

Hinweis:

Wenn Sie **Online Change** anwenden, können sich Inhalte von Adressen verschieben. Beachten Sie dies bei der Verwendung von Pointern auf Adressen!

7.7.3 BITADR

Operator: BITADR (bit address)

IEC-Operator: Nein, der Operator wird nicht von der Norm IEC 61131-3 gefordert.

Der Operator BITADR liefert den Bit-Offset innerhalb des Segments als DWORD. Beachten Sie, dass das Bit-Offset davon abhängt, ob in den Zielsystemeinstellungen die Option [Byteadressierung] aktiviert ist oder nicht.

Deklaration

```
VAR
   Var1 AT %IX2.3 : BOOL;
   bitoffset: DWORD;
END_VAR
```

Beispiel in AWL

```
LD Var1
BITADR
ST Var2
```

Beispiel in ST

```
bitoffset := BITADR(Var1); (* Ergebnis bei Byteadressierung=TRUE: 19,
                              bei Byteadressierung=FALSE: 35 *)
```

Hinweis:

Wenn Sie **Online Change** anwenden, können sich Inhalte von Adressen verschieben. Beachten Sie dies bei der Verwendung von Pointern auf Adressen!

7.7.4 Inhaltsoperator ^

Operator: ^

IEC-Operator: Ja.

Verwendung: Dereferenzierung von Pointern

Über den Inhaltsoperator "^" erfolgt die Dereferenzierung eines Pointers. Der Inhaltsoperator steht nach dem Pointerbezeichner.

Beispiel in ST

```
pt : POINTER TO INT;
Var_int1 : INT;
Var_int2 : INT;
pt := ADR(var_int1);
Var_int2 := pt^;
```

Hinweis:

Wenn Sie **Online Change** anwenden, können sich Inhalte von Adressen verschieben. Beachten Sie dies bei der Verwendung von Pointern auf Adressen!

7.8 Aufrufoperator

7.8.1 CAL

Operator: CAL

IEC-Operator: Ja.

Verwendung: Aufruf einer Funktionsblockinstanz

Mit dem Operator CAL rufen Sie in der Programmiersprache AWL eine Funktionsblockinstanz auf. Nach dem Namen der Funktinsblockinstanz folgt in runden Klammern die Belegung der Eingabevariablen des Funktionsblocks.

Beispiel

Aufruf der Instanz `Inst` eines Funktionsblocks mit Belegung der Eingabevariablen `Par1`, `Par2` auf 0 bzw. TRUE.

```
CAL INST(PAR1 := 0, PAR2 := TRUE)
```

7.9 Typkonvertierungen

7.9.1 Hinweise

Variablen können nicht implizit von einem "größeren" Typ zu einem "kleineren" Typ konvertiert werden, beispielsweise vom Typ `INT` zm Typ `BYTE`. Wenn Sie das tun möchten, müssen Sie spezielle Typkonvertierungen anwenden. Grundsätzlich können Sie Variablen von jedem elementaren Typ zu jedem anderen elementaren Typ konvertieren.

Syntax

```
<elem.Typ1>_TO_<elem.Typ2>
```

Hinweis:

Beachten Sie bei Konvertierungen `...TO_STRING`, dass der String linksbündig generiert wird. Wenn der String zu kurz definiert ist, wird rechts abgeschnitten.

Weitere Informationen finden Sie in den folgenden Kapiteln:

- (→ Kap. 7.9.2, BOOL_TO-Konvertierungen), Seite 361
- (→ Kap. 7.9.3, TO_BOOL-Konvertierungen), Seite 363
- (→ Kap. 7.9.4, Konvertierungen zwischen ganzzahligen Zahlentypen), Seite 364
- (→ Kap. 7.9.5, REAL_TO-/ LREAL_TO-Konvertierungen), Seite 365
- (→ Kap. 7.9.6, TIME_TO- / TIME_OF_DAY-Konvertierungen), Seite 366
- (→ Kap. 7.9.7, DATE_TO- / DT_TO-Konvertierungen), Seite 367
- (→ Kap. 7.9.8, STRING_TO-Konvertierungen), Seite 368
- (→ Kap. 7.9.9, TRUNC), Seite 369

7.9.2 BOOL_TO-Konvertierungen

Konvertierung vom Typ BOOL zu einem anderen Typ.

Bei Zahlentypen ist das Ergebnis 1, wenn der Operand TRUE ist und 0, wenn der Operand FALSE ist.

Beim Typ STRING ist das Ergebnis TRUE bzw. FALSE.

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

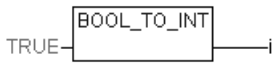
Beispiele in AWL

```
LD  TRUE          (* Ergebnis ist 1 *)
BOOL_TO_INT
ST  i

LD TRUE           (* Ergebnis ist 'TRUE' *)
BOOL_TO_STRING
ST str

LD TRUE           (* Ergebnis ist T#1ms *)
BOOL_TO_TIME
ST t

LD TRUE           (* Ergebnis ist TOD#00:00:00.001 *)
BOOL_TO_TOD
ST

LD FALSE          (* Ergebnis ist D#1970-01-01 *)
BOOL_TO_DATE
ST dat

LD TRUE           (* Ergebnis ist DT#1970-01-01-00:00:01 *)
BOOL_TO_DT
ST dandt
```

Beispiele in ST

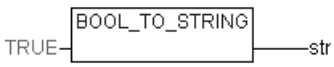
```
i := BOOL_TO_INT(TRUE);      (* Ergebnis ist 1 *)

str := BOOL_TO_STRING(TRUE); (* Ergebnis ist der String 'TRUE' *)

t := BOOL_TO_TIME(TRUE);     (* Ergebnis ist T#1ms *)

tof := BOOL_TO_TOD(TRUE);    (* Ergebnis ist TOD#00:00:00.001 *)

dat := BOOL_TO_DATE(FALSE);  (* Ergebnis ist D#1970-01-01 *)

dandt := BOOL_TO_DT(TRUE);   (* Ergebnis ist DT#1970-01-01-00:00:01 *)
```

Beispiele in FUP



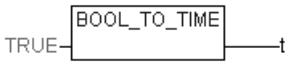
(* Ergebnis ist 1 *)

20628AXX



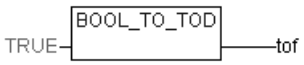
(* Ergebnis ist 'TRUE' *)

20629AXX



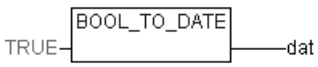
(* Ergebnis ist T#1ms *)

20630AXX



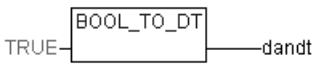
(* Ergebnis ist TOD#00:00:00.001 *)

20631AXX



(* Ergebnis ist D#1970-01-01 *)

20332AXX



(* Ergebnis ist DT#1970-01-01-00:00:01 *)

20633AXX

7.9.3 TO_BOOL-Konvertierungen

Konvertierung von einem Typ zum Typ BOOL.

Das Ergebnis ist TRUE, wenn der Operand ungleich 0 ist. Das Ergebnis ist FALSE, wenn der Operand gleich 0 ist.

Beim Typ STRING ist das Ergebnis TRUE, wenn der Operand TRUE ist, ansonsten ist das Ergebnis FALSE.

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

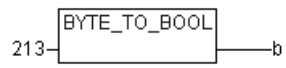
Beispiele in AWL

```
LD 213              (* Ergebnis ist TRUE *)
BYTE_TO_BOOL
ST b

LD 0                (* Ergebnis ist FALSE *)
INT_TO_BOOL
ST b

LD T#5ms            (* Ergebnis ist TRUE *)
TIME_TO_BOOL
ST b

LD 'TRUE'           (* Ergebnis ist TRUE *)
STRING_TO_BOOL
ST b
```

Beispiele in ST

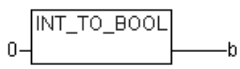
```
b := BYTE_TO_BOOL(2#11010101);  (* Ergebnis ist TRUE *)

b := INT_TO_BOOL(0);            (* Ergebnis ist FALSE *)

b := TIME_TO_BOOL(T#5ms);       (* Ergebnis ist TRUE *)

b := STRING_TO_BOOL('TRUE');    (* Ergebnis ist TRUE *)
```

Beispiele in FUP



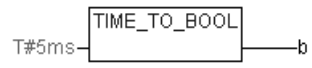
(* Ergebnis ist TRUE *)

20634AXX



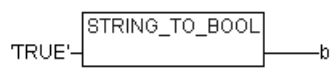
(* Ergebnis ist FALSE *)

20635AXX



(* Ergebnis ist TRUE *)

20636AXX



(* Ergebnis ist TRUE *)

20637AXX

7.9.4 Konvertierungen zwischen ganzzahligen Zahlentypen

Konvertierung von einem ganzzahligen Zahlentyp in einen anderen ganzzahligen Zahlentyp.

Bei der Typkonvertierung von größeren auf kleinere Typen können Informationen verloren gehen. Wenn die zu konvertierende Zahl die Bereichsgrenze überschreitet, dann werden die ersten Bytes der Zahl nicht berücksichtigt.

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

Beispiel in AWL

```
LD 2
INT_TO_REAL
MUL
```

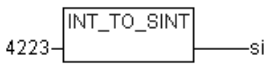
Beispiel in ST

```
si := INT_TO_SINT(4223); (* Ergebnis ist 127 *)
```

Hinweis:

Wenn Sie die Integer-Zahl 4223 (16#107f in Hexadezimaldarstellung) in eine Variable vom Typ `SINT` konvertieren, wird diese Zahl 127 (16#7f in Hexadezimaldarstellung).

Beispiel in FUP



20638AXX

7.9.5 REAL_TO-/ LREAL_TO-Konvertierungen

Konvertierung vom Typ REAL bzw. LREAL in einen anderen Typ.

Bei der Typkonvertierung wird nach oben oder unten auf einen ganzzahligen Wert gerundet und in den entsprechenden Typen gewandelt. Ausgenommen davon sind die Typen STRING, BOOL, REAL und LREAL.

Bei der Typkonvertierung von größeren auf kleinere Typen können Informationen verloren gehen.

Beachten Sie bei der Konvertierung in den Typ STRING, dass die Gesamtkommastellenzahl auf 16 begrenzt ist. Enthält die Zahl mehr Stellen, wird die 16. Stelle gerundet und so im String dargestellt. Wenn der String für die Zahl zu kurz definiert ist, wird rechts entsprechend abgeschnitten.

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

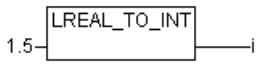
Beispiel in AWL

```
LD 2.7
REAL_TO_INT
GE %MW8
```

Beispiele in ST

```
i := REAL_TO_INT(1.5);    (* Ergebnis ist 2 *)

j := REAL_TO_INT(1.4);    (* Ergebnis ist 1 *)

i := REAL_TO_INT(-1.5);   (* Ergebnis ist -2 *)

j := REAL_TO_INT(-1.4);   (* Ergebnis ist -1 *)
```

Beispiel in FUP



20639AXX

7.9.6 TIME_TO- / TIME_OF_DAY-Konvertierungen

Konvertierung vom Typ TIME bzw. TIME_OF_DAY in einen anderen Typ.

Intern wird die Zeit im Format DWORD in Millisekunden abgespeichert. Beim Datentyp TIME_OF_DAY wird seit 00:00 Uhr gerechnet. Dieser DWORD-Wert wird konvertiert.

Bei der Typkonvertierung von größeren auf kleinere Typen können Informationen verloren gehen.

Beim Typ STRING ist das Ergebnis die Zeitkonstante.

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

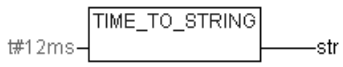
Beispiele in AWL

```
LD   T#12ms            (* Ergebnis ist 'T#12ms' *)
TIME_TO_STRING
ST   str

LD   T#300000ms
TIME_TO_DWORD          (* Ergebnis ist 300000 *)
ST   dw

LD   TOD#00:00:00.012  (* Ergebnis ist 12 *)
TOD_TO_SINT
ST   si
```

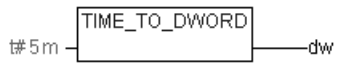
Beispiele in ST

```
str := TIME_TO_STRING(T#12ms);
dw := TIME_TO_DWORD(T#5m);
si := TOD_TO_SINT(TOD#00:00:00.012);
```

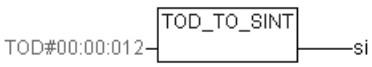
Beispiele in FUP



20640AXX



20641AXX



20642AXX

7.9.7 DATE_TO- / DT_TO-Konvertierungen

Konvertierung vom Typ DATE bzw. DATE_AND_TIME in einen anderen Typ.

Intern wird das Datum als DWORD in Sekunden seit dem 1. Januar 1970 abgespeichert. Dieser DWORD-Wert wird konvertiert.

Bei der Typkonvertierung von größeren auf kleinere Typen können Informationen verloren gehen.

Beim Typ STRING ist das Ergebnis die Datumskonstante.

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

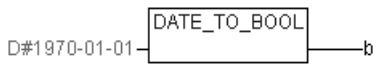
Beispiele in AWL

```
LD    D#1970-01-01              (* Ergebnis ist FALSE *)
DATE_TO_BOOL
ST    b

LD    D#1970-01-15              (* Ergebnis ist 29952 *)
DATE_TO_INT
ST    i

LD    DT#1970-01-15-05:05:05    (* Ergebnis ist 129 *)
DT_TO_BYTE
ST    byt

LD    DT#1998-02-13-14:20       (* Ergebnis ist 'DT#1998-02-13-14:20' *)
DT_TO STRING
ST    str
```

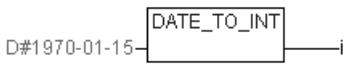
Beispiele in ST

```
b := DATE_TO_BOOL(D#1970-01-01);

i := DATE_TO_INT(D#1970-01-15);

byt := DT_TO_BYTE(DT#1970-01-15-05:05:05);

str := DT_TO_STRING(DT#1998-02-13-14:20);
```

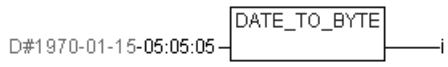
Beispiele in FUP



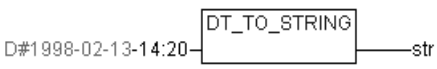
20643AXX



20644AXX



20645AXX



20646AXX

7.9.8 STRING_TO-Konvertierungen

Konvertierung vom Typ STRING in einen anderen Typ

Wenn der Operand vom Typ STRING keinen gültigen Wert des Zieltyps hat, ist das Ergebnis 0.

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

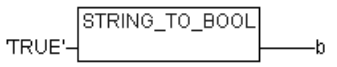
Beispiele in AWL

```
LD    'TRUE'            (* Ergebnis ist TRUE *)
STRING_TO_BOOL
ST    b

LD    'abc34'           (* Ergebnis ist 0 *)
STRING_TO_WORD
ST    w

LD    't#127ms'         (* Ergebnis ist T#127ms *)
STRING_TO_TIME
ST    t
```

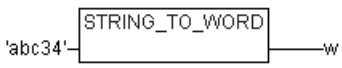
Beispiele in ST

```
b := STRING_TO_BOOL('TRUE');

w := STRING_TO_WORD('abc34');

t := STRING_TO_TIME('T#127ms');
```

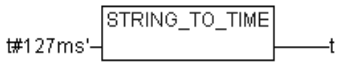
Beispiele in FUP



20647AXX



20648AXX



20649AXX

7.9.9 TRUNC

Konvertierung vom Typ `REAL` zum Typ `INT`.

Diese Konvertierung liefert den Betrag und das Vorzeichen vom ganzzahligen Anteil einer Zahl.

Bei der Typkonvertierung von größeren auf kleinere Typen können Informationen verloren gehen.

Beachten Sie bitte die Informationen auf Seite 346 (→ Kap. 7.5.1, Hinweise).

Beispiel in AWL

```
LD    2.7
TRUNC
GE    %MW8
```

Beispiele in ST

```
i := TRUNC(1.9); (* Ergebnis ist 1 *)
i := TRUNC(-1.4); (* Ergebnis ist -1 *)
```

7.10 Nummerische Operatoren

7.10.1 ABS

Operator: ABS

IEC-Operator: Ja

Verwendung: Dieser Operator liefert den Absolutwert einer Zahl.

Folgende Typkombinationen für den Eingang IN und den Ausgang OUT sind möglich:

IN	OUT
INT	INT, REAL, WORD, DWORD, DINT
REAL	REAL
BYTE	BYTE, WORD, DWORD, INT, DINT, REAL
WORD	WORD, DWORD, INT, DINT, REAL
DWORD	DWORD, DINT, REAL
SINT	REAL
USINT	REAL
UINT	WORD, DWORD, INT, DINT, UINT, UDINT, REAL
DINT	DWORD, DINT, REAL
UDINT	DWORD, DINT, UDINT, REAL

Beispiel in AWL

```
LD  -2
ABS
ST  i  (* Ergebnis ist 2 *)
```

Beispiel in ST

```
i := ABS(-2);
```

Beispiel in FUP

20650AXX

7.10.2 SQRT

Operator: SQRT

IEC-Operator: Ja

Verwendung: Dieser Operator liefert die Quadratwurzel einer Zahl.

Folgende Typen für den Eingang IN und den Ausgang OUT sind möglich:

IN: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

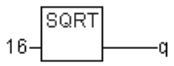
Beispiel in AWL

```
LD  16
SQRT
ST  q (* Ergebnis ist 4 *)
```

Beispiel in ST

```
q := SQRT(16);
```

Beispiel in FUP



20651AXX

7.10.3 LN

Operator: LN

IEC-Operator: Ja

Verwendung: Dieser Operator liefert den natürlichen Logarithmus einer Zahl.

Folgende Typen für den Eingang IN und den Ausgang OUT sind möglich:

IN: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

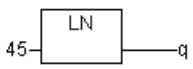
Beispiel in AWL

```
LD  45
LN
ST  q   (* Ergebnis ist 3.80666 *)
```

Beispiel in ST

```
q := LN(45);
```

Beispiel in FUP



20652AXX

7.10.4 LOG

Operator: LOG

IEC-Operator: Ja

Verwendung: Dieser Operator liefert den dekadischen Logarithmus (zur Basis 10) einer Zahl.

Folgende Typen für den Eingang IN und den Ausgang OUT sind möglich:

IN: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

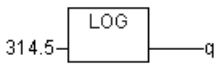
Beispiel in AWL

```
LD  314.5
LOG
ST  q   (* Ergebnis ist 2.49762 *)
```

Beispiel in ST

```
q := LOG(314.5);
```

Beispiel in FUP



20653AXX

7.10.5 EXP

Operator: EXP

IEC-Operator: Ja

Verwendung: Dieser Operator liefert den Exponentialwert einer Zahl.

Folgende Typen für den Eingang IN und den Ausgang OUT sind möglich:

IN: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

Beispiel in AWL

```
LD  2
EXP
ST  q (* Ergebnis ist 7.389056099 *)
```

Beispiel in ST

```
q := EXP(2);
```

Beispiel in FUP

20654AXX

7.10.6 SIN

Operator: SIN

IEC-Operator: Ja

Verwendung: Dieser Operator liefert den Sinus einer Zahl. Der Wert wird im Bogenmaß angegeben.

Folgende Typen für den Eingang IN und den Ausgang OUT sind möglich:

IN: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

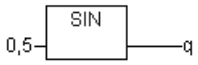
Beispiel in AWL

```
LD  0.5
SIN
ST  q  (* Ergebnis ist 0.479426 *)
```

Beispiel in ST

```
q := SIN(0.5);
```

Beispiel in FUP



20655AXX

7.10.7 COS

Operator: COS

IEC-Operator: Ja

Verwendung: Dieser Operator liefert den Cosinus einer Zahl. Der Wert wird im Bogenmaß angegeben.

Folgende Typen für den Eingang IN und den Ausgang OUT sind möglich:

IN: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

Beispiel in AWL

```
LD  0.5
COS
ST  q  (* Ergebnis ist 0.877583 *)
```

Beispiel in ST

```
q := COS(0.5);
```

Beispiel in FUP

20656AXX

7.10.8 TAN

Operator: TAN

IEC-Operator: Ja

Verwendung: Dieser Operator liefert den Tangens einer Zahl. Der Wert wird im Bogenmaß angegeben.

Folgende Typen für den Eingang IN und den Ausgang OUT sind möglich:

IN: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

Beispiel in AWL

```
LD  0.5
TAN
ST  q  (* Ergebnis ist 0.546302 *)
```

Beispiel in ST

```
q := TAN(0.5);
```

Beispiel in FUP

20657AXX

7.10.9 ASIN

Operator: ASIN

IEC-Operator: Ja

Verwendung: Dieser Operator liefert den Arcussinus (Umkehrfunktion des Sinus) einer Zahl. Der Wert wird im Bogenmaß angegeben.

Folgende Typen für den Eingang IN und den Ausgang OUT sind möglich:

IN: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

Beispiel in AWL

```
LD  0.5
ASIN
ST  q  (* Ergebnis ist 0.523599 *)
```

Beispiel in ST

```
q := ASIN(0.5);
```

Beispiel in FUP

20658AXX

7.10.10 ACOS

Operator: ACOS

IEC-Operator: Ja

Verwendung: Dieser Operator liefert den Arcuscosinus (Umkehrfunktion des Cosinus) einer Zahl. Der Wert wird im Bogenmaß angegeben.

Folgende Typen für den Eingang IN und den Ausgang OUT sind möglich:

IN: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

Beispiel in AWL

```
LD   0.5
ACOS
ST   q  (* Ergebnis ist 1.0472 *)
```

Beispiel in ST

```
q := ACOS(0.5);
```

Beispiel in FUP

20659AXX

7.10.11 ATAN

Operator: ATAN

IEC-Operator: Ja

Verwendung: Dieser Operator liefert den Arcustangens (Umkehrfunktion des Tangens) einer Zahl. Der Wert wird im Bogenmaß angegeben.

Folgende Typen für den Eingang IN und den Ausgang OUT sind möglich:

IN: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

Beispiel in AWL

```
LD   0.5
ATAN
ST   q  (* Ergebnis ist 0.463648 *)
```

Beispiel in ST

```
q := ATAN(0.5);
```

Beispiel in FUP

20660AXX

7.10.12 EXPT

Operator: EXPT

IEC-Operator: Ja

Verwendung: Potenzierung einer Zahl mit einer anderen Zahl.

Folgende Typen für die Eingänge IN1 bzw. IN2 und den Ausgang OUT sind möglich:

IN1, IN2: BYTE, WORD, DWORD, INT, DINT, SINT, USINT, UINT, UDINT, REAL

OUT: REAL

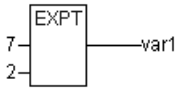
Beispiel in AWL

```
LD   7
EXPT 2
ST   var1 (* Ergebnis ist 49 *)
```

Beispiel in ST

```
var1 := EXPT(7,2);
```

Beispiel in FUP



20661AXX

7.11 Initialisierungsoperator

7.11.1 INI

Operator: `INI`

IEC-Operator: Ja

Verwendung: Mit dem Operator `INI` können `RETAIN`-Variablen einer im Baustein verwendeten Funktionsblockinstanz initialisiert werden.

Der Operator muss einer booleschen Variable zugewiesen werden.

Syntax

```
<bool-Variable> := INI(<FB-Instanz, TRUE|FALSE)
```

Wenn der zweite Parameter des Operators auf TRUE gesetzt ist, werden alle im Funktionsblock FB definierten `RETAIN`-Variablen initialisiert.

Beispiel in ST

`fbinst` ist die Instanz des Funktionsblocks `fb`, in dem eine Retain-Variable `retvar` definiert ist.

Deklaration im Baustein:

```
fbinst : fb;
b : bool;
```

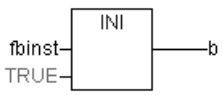
Programmteil:

```
b := INI(fbinst, TRUE);
ivar := fbinst.retvar  (* => retvar wird initialisiert *)
```

Beispiel des Operatoraufrufs in AWL

```
LD fbinst
INI TRUE
ST b
```

Beispiel des Operatoraufrufs in FUP



20662AXX

8 Operanden

Im PLC-Editor können Sie Konstanten, Variablen, Adressen und Funktionsaufrufe als Operanden verwenden.

8.1 Konstanten

8.1.1 Konstanten vom Typ BOOL

Konstanten vom Typ `BOOL` sind die Wahrheitswerte TRUE und FALSE.

8.1.2 Konstanten vom Typ TIME

In PLC-Editor können Sie Konstanten vom Typ `TIME` deklarieren, die insbesondere benutzt werden, um die Timer der Standardbibliothek zu bedienen.

Eine Konstante vom Typ `TIME` repräsentiert eine Zeitdauer.

Die Konstante besteht aus einem führenden `t` oder `time` bzw. `T` oder `TIME`, gefolgt von einem Doppelkreuz # und der Zeitdeklaration im Format:

- `d` Tage
- `h` Stunden
- `m` Minuten
- `s` Sekunden
- `ms` Millisekunden

Hinweis:

Die Zeitangaben müssen der Größe nach geordnet sein: d → h → m → s → ms.

Dabei müssen nicht alle Zeitangaben verwendet werden.

Beispiele für gültige TIME-Konstanten in einer ST-Zuweisung

```
TIME1 := T#14ms;
TIME1 := T#100s12ms; (*Überlauf in der höchsten Stelle ist erlaubt*)
TIME1 := t#12h34m15s;
```

Beispiele für ungültige TIME-Konstanten in einer ST-Zuweisung:

```
TIME1 := T#5m68s;    (*Überlauf in einer niedrigen Stelle ist verboten*)
TIME1 := 15ms;       (*Es fehlt T#*)
TIME1 := t#4ms13d;   (*Falsche Reihenfolge der Zeitangaben*)
```

8.1.3 Konstanten vom Typ DATE

Eine Konstante vom Typ DATE repräsentiert ein Kalenderdatum.

Die Konstante wird deklariert durch ein führendes d oder date bzw. D oder DATE, gefolgt von einem Doppelkreuz # und dem Datum in der Rehenfolge: Jahr-Monat-Tag.

Beispiele

```
DATE#1996-05-06
d#1972-03-29
```

Werte vom Typ DATE werden intern wie Daten vom Typ DWORD behandelt. Die Zeit wird in Sekunden ab dem 1. Januar 1970 00:00 Uhr berechnet.

8.1.4 Konstanten vom Typ TIME_OF_DAY

Eine Konstante vom Typ TIME_OF_DAY repräsentiert eine Uhrzeit.

Die Konstante wird deklariert durch ein führendes tod oder time_of_day bzw. TOD oder TIME_OF_DAY, gefolgt von einem Doppelkreuz # und der Zeit in der Schreibweise: Stunde:Minute:Sekunde.

Die Sekunden sind reelle Zahlen. Sie können auch Sekundenbruchteile angeben.

Beispiele

```
TIME_OF_DAY#15:36:30.123
tod#00:00:00
```

Werte vom Typ TIME_OF_DAY werden intern wie Daten vom Typ DWORD behandelt. Die Zeit wird in Millisekunden ab 00:00 Uhr berechnet.

8.1.5 Konstanten vom Typ DATE_AND_TIME

Eine Konstante vom Typ DATE_AND_TIME repräsentiert ein Datum und die Uhrzeit.

Sie wird deklariert durch ein anführendes dt oder date_and_time bzw. DT oder DATE_AND_TIME, gefolgt von einem Doppelkreuz #, dem Datum und der Zeit.

Das Datum und Zeit werden durch einen Bindestrich getrennt.

Beispiele

```
DATE_AND_TIME#1996-05-06-15:36:30
dt#1972-03-29-00:00:00
```

Werte vom Typ DATE_AND_TIME werden intern wie Daten vom Typ DWORD behandelt. Die Zeit wird in Sekunden ab dem 1. Januar 1970 00:00 Uhr berechnet.

8.1.6 Zahlenkonstanten

Zahlenwerte können als folgende Zahlentypen auftreten:

- Dualzahlen
- Oktalzahlen
- Dezimalzahlen
- Hexadezimalzahlen

Die Ziffernwerte für die Zahlen 10 bis 15 bei Hexadezimalzahlen werden wie üblich durch die Buchstaben A bis F angegeben.

Bei Dualzahlen, Oktalzahlen und Hexadezimalzahlen wird die Zahlenbasis, gefolgt von einem Doppelkreuz #, vor die Integerkonstante geschrieben.

Unterstriche innerhalb eines Zahlenwerts sind erlaubt.

Variablentypen der Zahlenwerte:

- `BYTE`
- `WORD`, `DWORD`
- `INT`, `SINT`, `USINT`, `UINT`, `DINT`, `UDINT`
- `REAL`

Beispiele

`14`	(Dezimalzahl)
`2#1001_0011`	(Dualzahl)
`8#67`	(Oktalzahl)
`16#A`	(Hexadezimalzahl)

Hinweis:

Implizite Konvertierungen von größeren in kleinere Datentypen sind nicht erlaubt. Beispielsweise kann eine Variable vom Typ `DINT` nicht ohne weiteres als Variable vom Typ `INT` benutzt werden. Benutzen Sie dafür die Typkonvertierungen.

8.1.7 Konstanten vom Typ REAL und LREAL

Konstanten vom Typ `REAL` und `LREAL` können Sie als Dezimalbruch und in Exponentionaldarstellung angeben. Benutzen Sie als Dezimaltrennzeichen kein Komma, sondern verwenden Sie die amerikanische Schreibweise mit einem Punkt als Dezimaltrennzeichen.

Beispiele für die Schreibweise von Konstanten vom Typ REAL und LREAL

`7.4`	(falsch: `7,4`)
`1.64e+009`	(falsch: `1,64e+009`)

Hinweis:

`LREAL`-Konstanten werden nicht unterstützt. `LREAL`-Konstanten können jedoch in `REAL`-Konstanten übersetzt werden.

Aktivieren Sie deshalb die entsprechende Option im Dialog [Ressourcen] / [Arbeitsbereich] / [Übersetzungsoptionen] ([Resources] / [Desktop] / [Build]).

8.1.8 Konstanten vom Typ STRING

Eine Konstante vom Typ STRING ist eine beliebige Zeichenkette variabler Länge. Konstanten vom Typ STRING werden mit einfachen Hochkommas vorn und hinten begrenzt. Sie können innerhalb eines Strings auch Leerzeichen und Umlaute eingeben. Leerzeichen und Umlaute werden genauso wie alle anderen Zeichen behandelt.

In Zeichenfolgen wird die Kombination des Dollarzeichens $ gefolgt von zwei hexadezimalen Ziffern als hexadezimale Darstellung des acht Bit ASCII-Zeichencodes interpretiert.

Kombinationen von Zeichen, die mit dem Dollarzeichen $ beginnen, werden wie folgt interpretiert:

$$	Dollarzeichen
$'	Hochkomma
$L oder $l	Zeilenvorschub
$N oder $n	Neue Zeile
$P oder $p	Seitenvorschub
$R oder $r	Zeilenumbruch
$T oder $t	Tabulator

Beispiele:

```
'w1Wüß?'
'Susi und Claus'
':-)'
'$FC'   => Ausgabe :ü
'firstline$Rsecondline'
```

8.1.9 Getypte Konstanten (Typed Literals)

Beim Rechnen mit IEC-Konstanten wird, mit Ausnahme von Konstanten vom Typ REAL, der kleinstmögliche Datentyp verwendet. Durch die Verwendung von getypten Konstanten (Typed Literals) kann ein anderer Datentyp verwendet werden, ohne dass die Konstante explizit deklariert werden muss. Die Konstante wird dabei mit einem Präfix versehen, das den Typ der Konstanten festlegt.

Syntax:

<Type>#<Literal>

Mit <Type> definieren Sie den gewünschten Datentyp.

Datentypen von <Type>:

- BOOL
- BYTE
- WORD, DWORD
- INT, SINT, USINT, UINT, DINT, UDINT
- REAL

Der Datentyp muss in Großbuchstaben geschrieben werden.

Mit <Literal> geben Sie eine Konstante an.

Der Wert der Konstanten muss zum Datentyp passen, der mit <Type> festgelegt ist.

Wenn die Konstante nicht ohne Datenverlust in den gewählten Datentyp überführt wer-

den kann, wird eine Fehlermeldung ausgegeben.

Verwenden Sie getypte Konstanten (Typed Literals) wie normale Konstanten.

Beispiel

```
Var1 := DINT#34;
```

8.2 Variablen

8.2.1 Variablen

Sie können Variablen entweder lokal im Deklarationsteil eines Bausteins oder in den globalen Variablenlisten deklarieren.

Hinweis:

Eine lokale Variable kann mit gleichem Namen definiert wie eine globale Variable. Innerhalb eines Bausteins hat die lokal definierte Variable stets Vorrang.

Zwei global definierte Variablen dürfen jedoch nicht den gleichen Namen haben.

Beispiel

Wenn Sie sowohl in einer globalen Variablenliste als auch in der Steuerungskonfiguration je eine Variable `var1` definieren, wird ein Übersetzungsfehler ausgegeben.

Für Variablen gelten folgende Regeln:

- Die Bezeichner dürfen keine Leerstellen und Umlaute enthalten.
- Die Bezeichner dürfen nicht doppelt deklariert werden.
- Die Bezeichner dürfen nicht identisch mit Schlüsselwörtern sein.
- Die Groß- und Kleinschreibung wird bei Variablen nicht beachtet.

  Beispiel: `VAR1`, `Var1` und `var1` sind keine unterschiedlichen Variablen.
- Unterstriche in den Bezeichnern sind signifikant.

  Beispiel: `A_BCD` und `AB_CD` werden als unterschiedliche Variablen interpretiert.
- Mehrfach aufeinanderfolgende Unterstriche am Anfang oder in einem Bezeichner sind nicht erlaubt.
- Die Bezeichnerlänge sowie der signifikante Bereich sind unbegrenzt.
- Variablen können überall verwendet werden, wo der deklarierte Datentyp es erlaubt.

Die verfügbaren Variablen stehen Ihnen im Dialog [Eingabehilfe] ([Help Manager]) zu Verfügung, den Sie mit der <F2>-Taste öffnen.

8.2.2 Systemflags

Systemflags sind implizit deklarierte Variablen, die von der Steuerung abhängig sind. Die Steuerung MOVI-PLC® verfügt über 1024 H-Variablen. Diese finden Sie in der Liste der globalen Variablen.

Um die Systemflags, die in der Steuerung zur Verfügung stehen, anzuschauen, führen Sie in Texteditoren den Menübefehl [Einfügen] / [Operand] ([Insert] / [Operand]) aus. Im Dialog [Eingabehilfe] wählen Sie die Kategorie [System Variable] ([System Variable]) aus.

8.2.3 Zugriff auf Variablen von Arrays, Strukturen und Bausteinen

Auf die Komponenten von zweidimensionalen Arrays greifen Sie mit folgender Syntax zu:

```
<Feld_Name>[Index1,Index2]
```

Auf die Variablen von Strukturen greifen Sie mit folgender Syntax zu:

```
<Strukturname>.<Variablenname>
```

Auf die Variablen von Funktionsblöcken und Programmen greifen Sie mit folgender Syntax zu:

```
<Bausteinname>.<Variablenname>
```

8.2.4 Adressierung von Bits in Variablen

In ganzzahligen Variablen können einzelne Bits angesprochen werden. Dazu hängen Sie den Index des zu adressierenden Bits an die Variable. Trennen Sie die Varable vom Index des Bits mit einem Punkt.

Der Bit-Index kann eine beliebige Konstante sein. Er muss jedoch kleiner als die Bit-Breite der Variablen sein.

Hinweis:

Die Indizierung der Variablen ist 0-basiert. Bei einer Indizierung mit dem Wert 2 wird das dritte Bit adressiert.

Beispiel

```
a : INT;
b : BOOL;
...
a.2 := b;
```

In diesem Beispiel wird das dritte Bit der Variablen a auf den Wert der Variablen b gesetzt.

Wenn der Index größer ist als die Bit-Breite der Variablen, wird folgender Fehler ausgegeben: "Index '<n>' außerhalb des gültigen Bereichs für Variable '<var>'".

Datentypen der Bitadressierung:

- `BYTE`
- `WORD`, `DWORD`
- `INT`, `SINT`, `USINT`, `UINT`, `DINT`, `UDINT`

Hinweis:

Ein Bit-Zugriff darf einer `VAR_IN_OUT`-Variablen nicht zugewiesen werden!

Bitzugriff mit Hilfe einer globalen Konstante

Sie können zur Indizierung des zu adressierenden Bits auch eine globale Konstante deklarieren.

Hinweis:

Um einen Bitzugriff über globale Konstanten durchführen zu können, muss die Option [Konstanten ersetzen] ([Replace constants]) im Menüpunkt [Projekt] / [Optionen] / [Übersetzungsoptionen] ([Project] / [Options] / [Build]) muss aktiviert sein!

Beispiel 1: Deklaration einer globalen Konstante in der globalen Variablenliste

```
VAR_GLOBAL CONSTANT
   enable : int := 2;
END_VAR
```

Beispiel 2: Bit-Adressierung bei einer ganzzahligen Variable:

Deklaration der Variablen im Baustein:

```
VAR
   xxx : int;
END_VAR
```

Bitzugriff:

```
xxx.enable := true;
```

Diese Anweisung setzt das dritte Bit der Variablen `xxx` auf TRUE.

Beispiel 3: Bit-Adressierung bei einer ganzzahligen Strukturkomponente

Deklaration der Struktur `stru1`:

```
TYPE stru1:
   STRUCT
      bvar : BOOL;
      rvar : REAL;
      wvar : WORD;
      {bitaccess enable 42 'Antrieb freigeben'}
   END_STRUCT
END_TYPE
```

Deklaration der Variablen im Baustein:

```
VAR
   x : stru1;
END_VAR
```

Bitzugriff:

```
xxx.enable := true;
```

Diese Anweisung setzt das 42. Bit der Variablen `x` auf TRUE.

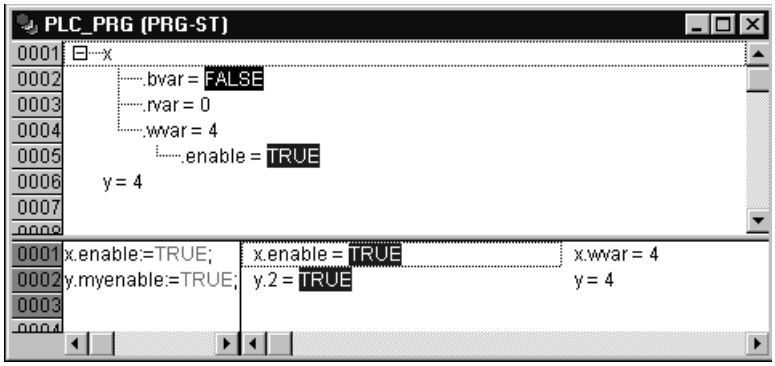
Da die Variable `bvar` eine Länge von 8 Bit und die Variable `rvar` eine Länge von 32 Bit hat, erfolgt die Adressierung des zweiten Bits der Variablen `wvar`.

Da das zweite Bit auf TRUE gesetzt wird, erhält die Variable `wvar` den Wert "4".

Hinweis:

Damit die Variable, die den Bitzugriff auf eine Strukturvariable mit Hilfe einer globalen Konstanten durchführt, korrekt dargestellt wird, verwenden Sie das Pragma `{bitacess}`. Informationen dazu finden Sie auf Seite 212 (→ Kap. 5.3.2, Pragmas für Initialisierung, Monitoring, Symbolerzeugung und für den Bit-Zugriff).

Die verwendeten globalen Konstanten werden dann beim Überwachen (Monitoring) im Deklarationseditor unterhalb der Strukturvariablen angezeigt.



20421AXX

8.3 Adressen

8.3.1 Hinweis

Hinweis:

Wenn Sie **Online Change** verwenden, können sich die Inhalte von Adressen verschieben.
Beachten Sie dies bei der Verwendung von Pointern auf Adressen.

8.3.2 Adressen

Die direkte Darstellung einzelner Speicherzellen erfolgt mit speziellen Zeichenreihen. Diese Zeichenreihen bestehen aus folgenden Angaben:

- einem Prozentzeichen %,
- einem Bereichspräfix,
- einem Präfix für die Größe
- einer oder mehreren natürlichen Zahlen, die durch Leerzeichen voneinander getrennt sind.

Folgende Bereichspräfixe werden unterstützt:

`I` Eingang

`Q` Ausgang

`M` Merker

Folgende Präfixe für die Größe werden unterstützt:

X	Einzelbit
None	Einzelbit
B	BYTE (8 Bit)
W	WORD (16 Bit)
D	DWORD (32 Bit)

Beispiele

%QX7.5 und %Q7.5	Ausgangsbit 7.5
%IW215	Eingangswort 215
%QB7	Ausgangsbyte 7
%MD48	Doppelwort an der Speicherstelle 48 im Merker
ivar AT %IW0:WORD;	Beispiel einer Variablendeklaration mit Adressangabe

Ob eine Adresse gültig ist, hängt von der aktuellen Steuerungskonfiguration des Programms ab.

Hinweis:

Boolesche Werte werden byteweise alloziert, wenn Sie nicht explizit eine Einzelbit-Adresse angeben.

Beispiel: Eine Wertänderung von varbool1 AT %QW0 betrifft den Bereich von QX0.0 bis QX0.7.

Hinweis:

Wenn Sie **Online Change** verwenden, können sich die Inhalte von Adressen verschieben.
Beachten Sie dies bei der Verwendung von Pointern auf Adressen.

8.3.3 Merker

Sie können alle unterstützten Größen für den Zugriff auf den Merker benutzen.

Beispiel

Die Adresse %MD48 adressiert die Bytes Nr. 192, 193, 194 und 195 im Merkerbereich (48 *4 = 192). Das erste Byte ist das Byte Nr. 0.

Sie können auch auf Worte, Bytes und Bits zugreifen.

Beispiel

Mit der Adresse %MX5.0 greifen Sie im fünften Wort auf das erste Bit zu. Bits werden in der Regel wortweise gespeichert.

Hinweis:

Wenn Sie **Online Change** verwenden, können sich die Inhalte von Adressen verschieben.
Beachten Sie dies bei der Verwendung von Pointern auf Adressen.

8.4 Funktionen

In der Programmiersprache ST kann auch ein Funktionsaufruf als Operand auftreten.

Beispiel:

```
Ergebnis := Fct(7) + 3;
```

TIME()-Funktion

Die Funktion `TIME()` liefert die seit dem Systemstart vergangene Zeit in Millisekunden. Der Datentyp ist `TIME`.

Beispiel in AWL

```
TIME
ST systime  (* Ergebnis z.B.: T#35m11s342ms *)
```

Beispiel in ST

```
systime := TIME();
```

Beispiel in FUP

20578AXX

9 Datentypen

9.1 Überblick

Sie können beim Programmieren Standard-Datentypen und selbst definierte Datentypen verwenden.

Jedem Bezeichner wird ein Datentyp zugeordnet, der festlegt, wieviel Speicherplatz reserviert wird und welche Werte dem Speicherinhalt entsprechen.

9.2 Standard-Datentypen

9.2.1 Datentyp BOOL

Variablen vom Datentyp BOOL können die Wahrheitswerte TRUE und FALSE annehmen. Es werden 8 Bit Speicherplatz reserviert.

9.2.2 Ganzzahlige Datentypen

Zu den ganzzahligen Datentypen zählen:

- BYTE
- WORD, DWORD
- INT, SINT, DINT, USINT, UINT, UDINT

Die verschiedenen Datenypen decken unterschiedliche Zahlenbereiche ab.

Für ganzzahlige Datentypen gelten die folgenden Bedingungen:

Datentyp	Bedeutung	Wertebereich	Speicherplatz
BYTE	Bitfolge von 8 Bit	0 ... 255	8 Bit
WORD	Bitfolge von 16 Bit	0 ... 65535	16 Bit
DWORD	Bitfolge von 32 Bit	0 ... 4294967295	32 Bit
SINT	Kurze ganze Zahl	-128 ... 127	8 Bit
INT	Ganze Zahl (Integer)	-32768 ... 32767	16 Bit
DINT	Doppelte ganze Zahl	-2147483648 ... 2147483647	32 Bit
USINT	Kurze ganze Zahl ohne Vorzeichen	0 ... 255	8 Bit
UINT	Ganze Zahl ohne Vorzeichen	0 ... 65535	16 Bit
UDINT	Doppelte ganze Zahl ohne Vorzeichen	0 ... 4294967295	32 Bit

Hinweis:

Bei der Typkonvertierung von größeren auf kleinere Datentypen können Informationen verloren gehen!

9.2.3 Datentyp REAL und LREAL

Die Datentypen REAL und LREAL sind sogenannte Gleitpunkttypen. Sie kommen bei der Verwendung von rationalen Zahlen zum Einsatz.

Der reservierte Speicherplatz beträgt:

- 32 Bit beim Typ REAL
- 64 Bit beim Typ LREAL

Zulässiger Wertebereich

REAL: 1.175494351e-38 ... 3.402823466e+38

LREAL: 2.2250738585072014e-308 ... 1.7976931348623158e+308

Hinweis:

Der Datentyp LREAL wird von der Steuerungskarte MOVI-PLC® DHP11B nicht unterstützt.

9.2.4 Datentyp STRING

Eine Variable vom Datentyp STRING ist eine beliebige Zeichenkette variabler Länge. Die Größenangabe zur Speicherplatzreservierung bei der Deklaration bezieht sich auf die Anzahl der Zeichen. Die Größenangabe erfolgt in runden oder eckigen Klammern. Wenn keine Größe angegeben ist, wird der Default-Initialwert von 80 Zeichen verwendet.

Die Anzahl der Zeichen in einem String ist grundsätzlich nicht begrenzt. Jedoch können String-Funktionen nur Strings bis 255 Zeichen verarbeiten!

Beispiel: Stringdeklaration mit 35 Zeichen

```
str : STRING(35) := 'Dies ist ein String';
```

9.2.5 Die Zeitdatentypen

Die folgenden Datentypen werden intern wie der Datentyp DWORD behandelt:

- TIME
- TIME_OF_DAY oder kurz TOD
- DATE und DATE_AND_TIME oder kurz DT

Bei den Datentypen TIME und TOD wird intern in Millisekunden gerechnet. Beim Datentyp TOD wird ab 00:00 Uhr gerechnet.

Bei den Datentypen DATE und DT wird intern in Sekunden ab dem 1. Januar 1970 um 00:00 Uhr gerechnet.

Beachten Sie die Zeitdatenformate der Zuweisung in den folgenden Abschnitten.

Konstanten vom Typ TIME

Eine Konstante vom Typ `TIME` repräsentiert eine Zeitdauer.

Die Konstante wird deklariert durch ein führendes `t` oder `time` bzw. `T` oder `TIME`, gefolgt von einem Doppelkreuz # und der Zeit im Format:

- `d` Tage
- `h` Stunden
- `m` Minuten
- `s` Sekunden
- `ms` Millisekunden

Maximaler Wert: 49d17h2m47s295ms. Das entspricht 4194967295 ms.

Hinweis:

Die Zeitangaben müssen der Größe nach geordnet sein: d → h → m → s → ms

Nicht alle Zeitangaben müssen verwendet werden.

Beispiele: gültige TIME-Konstanten in einer ST-Zuweisung

```
TIME1 := T#14ms;
TIME1 := T#100s12ms; (*Überlauf in der höchsten Stelle ist erlaubt*)
TIME1 := t#12h34m15s;
```

Beispiele: ungültige TIME-Konstanten in einer ST-Zuweisung

```
TIME1 := T#5m68s;    (*Überlauf in einer niedrigen Stelle verboten*)
TIME1 := 15ms;       (*Es fehlt T#*)
TIME1 := t#4ms13d;   (*Falsche Reihenfolge der Zeitangaben*)
```

Konstanten vom Typ DATE

Eine Konstante vom Typ `DATE` repräsentiert ein Kalenderdatum.

Sie wird deklariert durch ein anführendes `d` oder `date` bzw. `D` oder `DATE`, gefolgt von einem Doppelkreuz # und dem Datum in der Schreibweise: `Jahr-Monat-Tag`.

Möglicher Wertebereich: 1970-00-00 ... 2106-02-06

Beispiele

```
DATE#1996-05-06
d#1972-03-29
```

Konstanten vom Typ TIME_OF_DAY

Eine Konstante vom Typ `TIME_OF_DAY` repräsentiert eine Uhrzeit.

Sie wird deklariert durch ein anführendes `tod` oder `time_of_day` bzw. `TOD` oder `TIME_OF_DAY`, gefolgt von einem Doppelkreuz # und der Zeit in der Schreibweise: `Stunde:Minute:Sekunde`.

Die Sekunden sind reelle Zahlen. Sie können auch Sekundenbruchteile angeben.

Möglicher Wertebereich: 00:00:00 ... 1193:02:47.298

Beispiele

```
TIME_OF_DAY#15:36:30.123
tod#00:00:00
```

Konstanten vom Typ DATE_AND_TIME

Eine Konstante vom Typ DATE_AND_TIME repräsentiert ein Datum und die Uhrzeit.

Die Konstante wird deklariert durch ein anführendes dt oder date_and_time bzw. DT oder DATE_AND_TIME, gefolgt von einem Doppelkreuz #, dem Datum und der Zeit im Format: Jahr-Monat-Tag-Stunde:Minute:Sekunde.

Möglicher Wertebereich: 1970-00-00-00:00:00 ... 2106-02-06-06:28:15

Beispiele

```
DATE_AND_TIME#1996-05-06-15:36:30
dt#1972-03-29-00:00:00
```

9.3 Definierte Datentypen

9.3.1 Datentyp Array

Der PLC-Editor unterstützt ein-, zwei-, und dreidimensionale Arrays (Felder) von elementaren Datentypen. Die Arrays können Sie im Deklarationsteil eines Bausteins und in den globalen Variablenlisten definieren. Durch Schachtelung von Arrays dürfen maximal neun Dimensionen entstehen.

Syntax

<Feld_Name>:ARRAY [<ug1>..<og1>,<ug2>..<og2>] OF <elem. Typ>;

Die Werte ug1, ug2 definieren die untere Grenze des Feldbereichs.

Die Werte og1, og2 definieren die obere Grenze des Feldbereichs.

Diese Grenzwerte müssen ganzzahlig sein und dem Wertebereich des Datentyps DINT entsprechen.

Beispiel

```
Kartenspiel : ARRAY [1..13, 1..4] OF INT;
```

Initialisierung von Arrays

Beispiele: Komplette Initialisierung eines Arrays

```
arr1 : ARRAY [1..5] OF INT := 1,2,3,4,5;
arr2 : ARRAY [1..2,3..4] OF INT := 1,3(7);
             (* 1,3(7) ist kurz für 1,7,7,7 *)
arr3 : ARRAY [1..2,2..3,3..4] OF INT := 2(0),4(4),2,3;
             (* 2(0),4(4),2,3 ist kurz für 0,0,4,4,4,4,2,3 *)
```

Beispiel: Initialisierung des Arrays einer Struktur

```
TYPE STRUCT1
STRUCT
   p1 : int;
   p2 : int;
   p3 : dword;
END_STRUCT

ARRAY[1..3] OF STRUCT1 := (p1:=1,p2:=10,p3:=4723),
   (p1:=2,p2:=0,p3:=299),(p1:=14,p2:=5,p3:=112);
```

Beispiel: Teilweise Initialisierung eines Arrays

```
arr1 : ARRAY [1..10] OF INT := 1,2;
```

Feldelemente für die kein Wert vorgegeben wird, werden mit dem Default-Initialwert des Datentypen initialisiert. Im obigen Beispiel werden also die Feldelemente ARRAY [3] bis ARRAY [10] mit "0" initialisiert.

Zugriff auf Array-Komponenten

Syntax für den Zugriff auf Feldkomponenten bei einem zweidimensionalen Array:

```
<Feld_Name>[Index1,Index2]
```

Beispiel

```
Kartenspiel[9,2]
```

Hinweis:

Wenn Sie im Projekt eine Funktion mit Namen CheckBounds definieren, können Sie mit dieser Funktion Bereichsüberschreitungen bei Arrays automatisch überprüfen!

Funktion CheckBounds

Mit der Funktion CheckBounds können Sie in einem Projekt Bereichsüberschreitungen in Arrays automatisch überprüfen. Der Name der Funktion ist festgelegt und darf nur diese Bezeichnung besitzen.

Beispiel: Funktion CheckBounds

```
FUNCTION CheckBounds : DINT
   VAR_INPUT
      index, lower, upper: DINT;
   END_VAR
   IF index < lower THEN
      CheckBounds := lower;
   ELSIF index > upper THEN
        CheckBounds := upper;
      ELSE CheckBounds := index;
   END_IF
```

Test-Programm für die Funktion CheckBounds:

```
PROGRAM PLC_PRG
   VAR
      a : ARRAY[0..7] OF BOOL;
      b : INT := 10;
   END_VAR
   a[b] := TRUE;
```

Dieses Beispielprogramm dient zum Testen der Funktion CheckBounds. Das Programm greift auf Werte außerhalb der Grenzen eines definierten Arrays zu. Die Funktion CheckBounds gewährleistet, dass der Wert TRUE nicht an die Stelle a[10], sondern an der oberen noch gültigen Bereichsgrenze a[7] zugewiesen wird. Die Funktion CheckBounds korrigiert somit die Zugriffe auf Daten außerhalb der Array-Grenzen.

Hinweis:

Die in der Bibliothek `check.lib` enthaltene Funktion `CheckBounds` ist eine Beispiellösung!

Prüfen Sie vor Verwendung dieser Bibliothek, ob die Funktion wie gewünscht arbeitet oder implementieren Sie eine entsprechende Funktion `CheckBounds` als Baustein direkt im Projekt.

9.3.2 Pointer

Ein Pointer dient zur Speicherung der Adresse einer Variable oder eines Funktionsblocks in der Laufzeit eines Programms.

Syntax der Pointerdeklaration

```
<Bezeichner> : POINTER TO <Datentyp/Funktionsblock>;
```

Ein Pointer kann auf jeden beliebigen Datentyp und Funktionsbaustein zeigen, auch auf einen selbst definierten Datentyp oder Funktionsbaustein.

Mit dem Adressoperator `ADR` weisen Sie dem Pointer die Adresse einer Variablen oder Funktionsbausteins zu.

Die Dereferenzierung eines Pointers erfolgt mit dem Inhaltsoperator `^`, der nach dem Pointerbezeichner steht.

Beispiel

```
pt : POINTER TO INT;
var_int1 : INT := 5;
var_int2 : INT;
pt := ADR(var_int1);
var_int2 := pt^; (* var_int2 ist nun 5 *)
```

Hinweis:

Ein Pointer wird byte-weise hochgezählt!

Mit der Anweisung `p = p + SIZEOF(p^);` erreichen Sie das Hochzählen wie im C-Compiler.

Funktion CheckPointer

Durch Implementierung der Funktion `CheckPointer` können Sie überprüfen, ob die vom Pointer angesprochene Adresse im gültigen Speicherbereich liegt.

Die Funktion `CheckPointer` wird vor jedem Zugriff auf den Inhalt eines Pointers automatisch aufgerufen.

Die Funktion `CheckPointer` muss direkt im Projekt oder über eine Bibliothek verfügbar sein.

Beispiel: Systeme, die 32-Bit-Pointer verwenden

```
FUNCTION CheckPointer : DWORD
   VAR_INPUT
      dwAddress : DWORD;
      iSize : INT;
      bWrite : BOOL;
   END_VAR
```

Beispiel: Systeme, die 16-Bit-Pointer verwenden

```
FUNCTION CheckPointer : WORD
   VAR_INPUT
      dwAddress : WORD;
      iSize : INT;
      bWrite : BOOL;
   END_VAR
```

Die Funktion `CheckPointer` liefert die Adresse, die für die Dereferenzierung des Pointers verwendet wird. Wenn die Adresse im gültigen Speicherbereich liegt wird die Adresse geliefert, die als Eingangsparameter `dwAddress` übergeben wurde.

Hinweis:

Wenn Sie **Online Change** verwenden, können sich die Inhalte von Adressen verschieben.
Beachten Sie dies bei der Verwendung von Pointern auf Adressen.

9.3.3 Aufzählungstyp, Enumeration

Ein Aufzählungstyp ist ein selbstdefinierter Datentyp, der aus einer Menge von String-Konstanten besteht. Diese Konstanten sind die Enumerationswerte.

Die Enumerationswerte sind im ganzen Projekt bekannt, auch wenn sie lokal in einem Baustein deklariert wurden. Legen Sie die Aufzählungstypen stets als Objekte im Object Organizer in der Registerkarte Datentypen ([Data types]) an.

Aufzählungstypen beginnen mit dem Schlüsselwort `TYPE` und enden mit dem Schlüsselwort `END_TYPE`.

Syntax

```
TYPE <Bezeichner> : (<Enum_0> ,<Enum_1>, ...,<Enum_n>);
END_TYPE
```

Die Variable vom Typ `<Bezeichner>` kann einen der Enumerationswerte annehmen und wird mit dem ersten Enumerationswert initialisiert.

Die Werte sind zu ganzen Zahlen kompatibel. Sie können damit Operationen wie mit dem Datentyp `INT` durchführen. Einer Variablen können Sie eine Zahl x zuweisen. Wenn die Enumerationswerte nicht initialisiert sind, beginnt die Zählung bei 0. Achten Sie beim Initialisieren darauf, dass die Initialwerte aufsteigend sind. Die Gültigkeit der Zahl wird zur Laufzeit überprüft.

Beispiel

```
TYPE AMPEL : (Rot, Gelb, Gruen:=10);
          (* Rot hat den Initalwert 0, Gelb 1, Gruen 10 *)
END_TYPE

AMPEL1 : AMPEL;
AMPEL1 := 0; (* Ampel hat den Wert Rot *)
FOR i := Rot TO Gruen DO
   i := i + 1;
END_FOR;
```

Der gleiche Enumerationswert darf sowohl innerhalb einer Enumeration als auch bei der Verwendung verschiedener Enumerationen innerhalb des selben Bausteins nicht zweimal verwendet werden.

Beispiel

```
AMPEL : (rot, gelb, gruen);
FARBE : (blau, weiss, rot);
```

Fehler: Der Enumerationswert `rot` darf nicht für die Variablen AMPEL und FARBE verwendet werden, wenn diese im gleichen Baustein verwendet werden.

Hinweis:

Die Enumerationswerte dürfen nicht mit den nachfolgend aufgeführten Zeichenfolgen beginnen:

- MDX_
- MX_
- MC07_
- MM_

9.3.4 Strukturen

Legen Sie Strukturen als Objekte in der Registerkarte [Datentypen] ([Data types]) des Object Organizers an.

Strukturen beginnen mit den Schlüsselwörtern TYPE und STRUCT und enden mit den Schlüsselwörtern END_STRUCT und END_TYPE.

Syntax

```
TYPE <Strukturname>:
   STRUCT
      <Variablendeklaration 1>
      ...
      <Variablendeklaration n>
   END_STRUCT
END_TYPE
```

<Strukturname> ist ein Datentyp, der im gesamten Projekt bekannt ist. <Strukturname> können Sie wie einen Standard-Datentypen verwenden.

Verschachtelte Strukturen sind erlaubt. Die einzige Einschränkung ist, dass die Variablen nicht auf Adressen gesetzt werden können. Die AT-Deklaration ist also nicht erlaubt!.

Beispiel: Strukturdefinition

```
TYPE Polygonzug:
   STRUCT
      Start : ARRAY [1..2] OF INT;
      Punkt1 : ARRAY [1..2] OF INT;
      Punkt2 : ARRAY [1..2] OF INT;
      Punkt3 : ARRAY [1..2] OF INT;
      Punkt4 : ARRAY [1..2] OF INT;
      Ende : ARRAY [1..2] OF INT;
   END_STRUCT
END_TYPE
```

Beispiel: Initialisierung dieser Struktur

```
Poly_1 : Polygonzug := (Start := 3,3, Punkt1 := 5,2, Punkt2 := 7,3,
Punkt3 := 8,5, Punkt4 := 5,7, Ende := 3,5);
```

Initialisierungen mit Variablen sind nicht möglich. Ein Beispiel für die Initialisierung eines Arrays einer Struktur finden Sie auf Seite 392 (→ Kap. 9.3.1, Datentyp Array).

Zugriff auf Strukturen

Syntax für den Zugriff auf Komponenten einer Struktur:

```
<Struktur_Name>.<Komponentenname>
```

Beispiel

Für das oben genannte Beispiel der Struktur `Polygonzug` erfolgt der Zugriff auf die Komponente `Start` dementsprechend mit `Poly_1.Start` .

9.3.5 Referenzen

Mit Referenzen können Sie einen alternativen Namen (Alias) für einen Datentyp oder einen Funktionsblock festlegen.

Legen Sie Referenzen als Objekte in der Registerkarte [Datentypen] ([Data types]) des **Object Organizer** an.

Referenzen beginnen mit dem Schlüsselwort `TYPE` und enden mit dem Schlüsselwort `END_TYPE`.

Syntax

```
TYPE <Bezeichner>:<Zuweisungsausdruck>;
END_TYPE
```

Beispiel

```
TYPE message:STRING[50];
END_TYPE
```

9.3.6 Unterbereichstypen

Ein Unterbereichstyp ist ein Datentyp, dessen Wertebereich nur eine Untermenge eines bestimmten Datentyps umfasst.

Die Deklaration eines Unterbereichstyps nehmen Sie in der Registerkarte [Datentypen] ([Data types]) vor. Außerdem können Sie eine Variable direkt mit einem Unterbereichstypen deklarieren.

Syntax für die Deklaration im Registerblatt [Datentypen] ([Data types])

```
TYPE <Name> : <Inttype> (<ug>..<og>);
END_TYPE
```

`<Name>` muss ein gültiger IEC-Bezeichner sein.

Datentypen von `<Inttype>`:

- `BYTE`
- `WORD`, `DWORD`, (`LWORD`)
- `INT`, `SINT`, `USINT`, `UINT`, `DINT`, `UDINT`, (`LINT`, `ULINT`)
- `REAL`

`<ug>` ist eine Konstante, die die Untergrenze des Wertebereichs des Unterbereichstyps festlegt. Die Untergrenze `<ug>` liegt innerhalb des Wertebereichs.

`<og>` ist eine Konstante, die die Obergrenze des Wertebereichs des Unterbereichstyps festlegt. Die Obergrenze `<og>` liegt innerhalb des Wertebereichs.

Beispiel

```
TYPE SubInt : INT (-4095..4095);
END_TYPE
```

Beispiel: Direkte Deklaration einer Variablen mit einem Unterbereichstyp

Wenn der Wertebereich des Unterbereichstyps nicht den Wert "0" enthält, müssen Sie einen Initialwert angeben.

```
VAR
   i1 : INT (-4095..4095);
   i2 : INT (5..10) := 5; (* 5 ist der Initialwert *)
   ui : UINT (0..10000);
END_VAR
```

Wenn Sie in der Deklaration oder in der Implementation einem Unterbereichstyp eine Konstante zuweisen, die nicht im Wertebereich liegt (z. B. `i := 5000`), wird eine Fehlermeldung ausgegeben.

Funktionen CheckRangeSigned und CheckRangeUnsingned

Um die Einhaltung der Bereichsgrenzen im Online-Modus zu überprüfen, fügen Sie die Funktionen `CheckRangeSigned` bzw. `CheckRangeUnsigned` ein.

- Verwenden Sie die Funktion `CheckRangeSigned`, wenn der Datentyp des Unterbereichstypen mit Vorzeichen ist (z. B. `INT`, `DINT`). Informationen dazu finden Sie auf Seite 389 (→ Kap. 9.2.2, Ganzzahlige Datentypen).
- Verwenden Sie die Funktion `CheckRangeUnsigned` wenn der Datentyp des Unterbereichstypen ohne Vorzeichen ist (z. B. `UINT`, `UDINT`). Informationen dazu finden Sie auf Seite 389 (→ Kap. 9.2.2, Ganzzahlige Datentypen).

In diesen Funktionen können Sie Bereichsverletzungen in geeigneter Art und Weise abfangen. Sie können beispielsweise den Wert abschneiden oder einen Fehlerflag setzen.

Die Funktionen `CheckRangeSigned` bzw. `CheckRangeUnsigned` werden implizit aufgerufen, wenn auf eine Variable geschrieben wird, die von einem Unterbereichstyp ist, der aus einem Typ mit bzw. ohne Vorzeichen gebildet wurde.

Syntax der Funktion CheckRangeSigned

```
FUNCTION CheckRangeSigned : DINT
   VAR_INPUT
      value, lower, upper: DINT;
   END_VAR
```

Syntax der Funktion CheckRangeUnsigned

```
FUNCTION CheckRangeUnsigned : UDINT
   VAR_INPUT
      value, lower, upper: UDINT;
   END_VAR
```

Beispiel: Funktion CheckRangeSigned

Die Funktion `CheckRangeSigned` kann folgendermaßen programmiert werden, um einen Wert auf den erlaubten Bereich zurückzuschneiden:

```
FUNCTION CheckRangeSigned : DINT
   VAR_INPUT
      value, lower, upper : DINT;
   END_VAR
   IF (value < lower) THEN
      CheckRangeSigned := lower;
   ELSIF(value > upper) THEN
        CheckRangeSigned := upper;
      ELSE
        CheckRangeSigned := value;
   END_IF
```

Für einen automatischen Aufruf sind der Funktionsname `CheckRangeSigned` bzw. `CheckRangeUnsigned` und der Aufbau der Schnittstelle aus einem Rückgabewert und drei Parametern vom Typ `DINT` zwingend erforderlich.

Die Funktion wird beim Aufruf folgendermaßen parametriert:

- `value` erhält den Wert, der dem Bereichstypen zugewiesen werden soll.
- `lower` erhält die Untergrenze des Bereichs.
- `upper` erhält die Obergrenze des Bereichs.
- `Return value` erhält den Wert, der tatsächlich dem Bereichstyp zugewiesen wird.

Aus der Zuweisung `i := 10*y;` wird in diesem Beispiel implizit folgende Zuweisung erzeugt:

```
i := CheckRangeSigned(10*y, -4095, 4095);
```

Wenn `y` beispielsweise den Wert 1000 hat, dann hat `i` nach dieser Zuweisung trotzdem nur den Wert 4095.

Hinweis:

Wenn die beiden Funktionen `CheckRangeSigned` und `CheckRangeUnsigned` nicht vorhanden sind, findet im Online-Modus keine Typüberprüfung der Unterbereichstypen statt! Die Variable `i` könnte dann durchaus beliebige Werte zwischen -32768 und 32767 annehmen!

Hinweis:

Wenn die Funktion `CheckRangeSigned` bzw. `CheckRangeUnsigned` implementiert ist, kann bei der Verwendung des Unterbereichstypen in einer `FOR`-Schleife eine Endlosschleife entstehen. Dies geschieht dann, wenn der für die `FOR`-Schleife angegebene Bereich gleich groß, oder größer ist als der Bereich des Unterbereichstyps!

Hinweis:

Die in der Bibliothek `check.lib` enthaltene Funktion `CheckRangeSigned` ist eine Beispiellösung! Prüfen Sie vor Verwendung dieser Bibliothek, ob die Funktion wie gewünscht arbeitet oder implementieren Sie die entsprechende Funktion `CheckRange` direkt als Baustein im Projekt.

Beispiel

```
VAR
   ui : UINT (0..10000);
END_VAR
FOR ui := 0 TO 10000 DO
   ...
END_FOR
```

Die FOR-Schleife wird nicht verlassen, weil ui nicht größer als 10000 werden kann.

Ebenso ist der Inhalt der Funktionen CheckRangeSigned und CheckRangeSigned bei der Verwendung von Inkrementationswerten in der FOR-Schleife zu beachten!

Hinweis:

Beachten Sie ebenso den Inhalt der Funktionen CheckRangeSigned und CheckRangeSigned bei der Verwendung von Inkrementationswerten in der FOR-Schleife!

10 Bibliotheken

10.1 Übersicht der Bibliotheken

Im PLC-Editor können Sie zur Nutzung von Standardbausteinen der Norm IEC 61131-3 die Bibliothek `standard.lib` einbinden. Darüberhinaus können Sie im PLC-Editor zahlreiche weitere Bibliotheken zur optimierten Ansteuerung von Antriebs- und Frequenzumrichtern von SEW sowie weitere Peripheriemodule einbinden.

Die folgende Übersicht zeigt die Grundbibliotheken zur Ansteuerung von Geräten, die an die Steuerung MOVI-PLC® angeschlossen werden. Zusätzlich zu diesen Bibliotheken stehen Ihnen je nach Umrichtertyp weitere applikationsspezifische Bibliotheken für Handling, Kurvenscheiben, Synchronlauf, Wickelanwendungen usw. zur Verfügung.

MPLCProcessdata	**MPLCMotion_MDX**	**MPLCMotion_MC07**	**MPLCMotion_MX**	**MPLCMotion_MM**	**MPLCUtilities**
MOVI-PLC® wie herkömmliche Steuerung nutzbar • Ansteuerung aller SEW-Umrichter über Prozessdaten • für Nutzung von Prozessdatenprofilen, Applikationsmodulen oder eigenen IPOS® Programmen	MOVI-PLC® als Motion Controller für MOVIDRIVE® B • Einachs-Bewegungsbefehle • Nutzung der Schnittstellen von MOVIDRIVE® B	MOVI-PLC® als Motion Controller für MOVITRAC® 07 • Drehzahlbefehle • Nutzung der Schnittstellen von MOVITRAC® 07	MOVI-PLC® als Motion Controller für MOVIAXIS® • Einachs-Bewegungsbefehle • Nutzung der Schnittstellen von MOVIAXIS®	MOVI-PLC® als Motion Controller für MOVIMOT® • Drehzahlbefehle • Nutzung der Schnittstellen von MOVIMOT®	Anschluss von z. B. CANopen IO-Modulen
↓	↓	↓	↓	↓	↓
SEW Processdata Module	MOVIDRIVE® MDX	MOVITRAC® 07	MOVIAXIS®	MOVIMOT®	CANopen IO-Module
Elemente, die zur Nutzung der Bibliotheken in der Steuerungskonfiguration des PLC-Editors eingebunden werden müssen					

Übersicht der umrichter-/gerätespezifischen Bibliotheken für Motion- und Ein-/Ausgänge

Im vorliegenden Programmierhandbuch sind die Bibliotheken `standard.lib` und `MPLProcessdata.lib` beschrieben. Die weiteren Bibliotheken sind in separaten Handbüchern mit der Bezeichnung "<Bibliotheksname> für MOVI-PLC®" dokumentiert.

10.2 Bibliothek standard.lib

Die Bibliothek `standard.lib` steht Ihnen standardmäßig zur Verfügung. Sie enthält alle Funktionen und Funktionsbausteine, die von der Norm IEC 61131-3 als Standardbausteine für ein IEC-Programmiersystem gefordert werden. Der Unterschied zwischen einer Standardfunktion und einem Operator besteht darin, dass der Operator dem Programmiersystem implizit bekannt ist, während Standardbausteine mit der Bibliothek `standard.lib` an das Projekt gebunden werden müssen.

10.2.1 String-Funktionen

Achtung:

String-Funktionen sind nicht "thread-safe".

Bei der Verwendung von Tasks dürfen Stringfunktionen nur in einer Task verwendet werden! Wenn Sie die gleiche Stringfunktion in verschiedenen Tasks verwenden, besteht bei der Abarbeitung der Tasks die Gefahr des Überschreibens.

Die zulässige String-Länge beim Verwenden der Funktionen ist 1 bis 255.

CONCAT

Die Funktion `CONCAT` der Bibliothek `standard.lib` liefert die Konkatenation (Aneinanderreihung) von zwei Strings.

Die Eingänge `STR1` und `STR2` sind vom Typ `STRING`.

Der Ausgang der Funktion ist vom Typ `STRING`.

`CONCAT (STR1, STR2)` bedeutet: Liefere die Konkatenation von `STR1` und `STR2`.

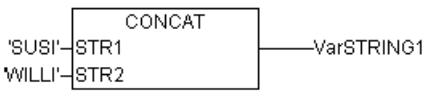
Beispiel in AWL

```
LD      'SUSI'
CONCAT  'WILLI'
ST      VarSTRING1 (* Ergebnis ist 'SUSIWILLI' *)
```

Beispiel in ST

```
VarSTRING1 := CONCAT ('SUSI','WILLI');
```

Beispiel in FUP



20554AXX

Achtung:

String-Funktionen sind nicht "thread-safe"!

DELETE

Die Funktion DELETE der Bibliothek standard.lib löscht einen Teilstring ab einer bestimmten Position aus einem String.

Der Eingang STR ist vom Typ STRING.

Die Eingänge LEN und POS sind vom Typ INT.

Der Ausgang der Funktion ist vom Typ STRING.

DELETE (STR, LEN, POS) bedeutet: Lösche LEN Zeichen aus STR, beginnend an der Position POS.

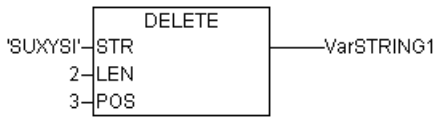
Beispiel in AWL

```
LD      'SUXYSI'
DELETE  2,3
ST      VarSTRING1 (* Ergebnis ist 'SUSI' *)
```

Beispiel in ST

```
VarSTRING1 := DELETE ('SUXYSI',2,3);
```

Beispiel in FUP



20556AXX

Achtung:

String-Funktionen sind nicht "thread-safe"!

FIND

Die Funktion FIND der Bibliothek standard.lib sucht einen Teilstring in einem String.

Die Eingänge STR1 und STR2 sind vom Typ STRING.

Der Ausgang der Funktion ist vom Typ INT.

FIND (STR1, STR2) bedeutet: Finde die Position des ersten Zeichens des ersten Vorkommens von STR2 in STR1. Wenn STR2 in STR1 nicht enthalten ist, dann ist der Ausgang der Funktion := 0.

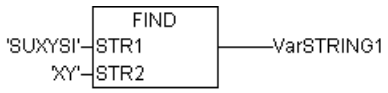
Beispiel in AWL

```
LD   'SUXYSI'
FIND 'XY'
ST   VarINT1 (* Ergebnis ist '3' *)
```

Beispiel in ST

```
VarINT1 := FIND ('SUXYSI','XY');
```

Beispiel in FUP



20558AXX

Achtung:

String-Funktionen sind nicht "thread-safe"!

INSERT

Die Funktion `INSERT` der Bibliothek `standard.lib` fügt einen String ab einer bestimmten Position in einen anderen String ein.

Die Eingänge `STR1` und `STR2` sind vom Typ `STRING`.

Der Eingang `POS` ist vom Typ `INT`.

Der Ausgang der Funktion ist vom Typ `STRING`.

`INSERT (STR1, STR2, POS)` bedeutet: Füge `STR2` in `STR1` ab der Position `POS` ein.

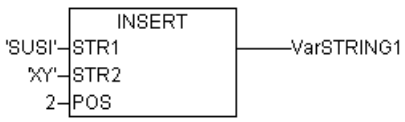
Beispiel in AWL

```
LD      'SUSI'
INSERT  'XY',2
ST      VarSTRING1 (* Ergebnis ist 'SUXYSI' *)
```

Beispiel in ST

```
VarSTRING1 := INSERT ('SUSI','XY',2);
```

Beispiel in FUP



20555AXX

Achtung:

String-Funktionen sind nicht "thread-safe"!

LEN

Die Funktion LEN der Bibliothek standard.lib liefert die Länge eines Strings.

Der Eingang STR der Funktion ist vom Typ STRING.

Der Ausgang der Funktion ist vom Typ INT.

LEN(STR) bedeutet: Liefere die Anzahl der Zeichen des Strings STR.

Beispiel in AWL

```
LD   'SUSI'
LEN
ST   VarINT1 (* Ergebnis ist 4 *)
```

Beispiel in ST

```
VarSTRING1 := LEN ('SUSI');
```

Beispiel in FUP

20550AXX

Achtung:

String-Funktionen sind nicht "thread-safe"!

LEFT

Die Funktion LEFT der Bibliothek standard.lib liefert den linken Anfangsstring eines Strings.

Der Eingang STR ist vom Typ STRING.

Der Eingang SIZE ist vom Typ INT.

Der Ausgang der Funktion ist vom Typ STRING.

LEFT (STR, SIZE) bedeutet: Liefere die ersten SIZE Zeichen von links im String STR.

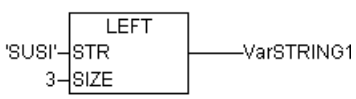
Beispiel in AWL

```
LD   'SUSI'
LEFT 3
ST   VarSTRING1 (* Ergebnis ist 'SUS' *)
```

Beispiel in ST

```
VarSTRING1 := LEFT ('SUSI',3);
```

Beispiel in FUP



20551AXX

Achtung:
String-Funktionen sind nicht "thread-safe"!

MID

Die Funktion MID der Bibliothek standard.lib liefert einen Teilstring eines Strings.

Der Eingang STR ist vom Typ STRING.

Die Eingänge LEN und POS sind vom Typ INT.

Der Ausgang der Funktion ist vom Typ STRING.

MID (STR, LEN, POS) bedeutet: Liefere LEN Zeichen aus dem String STR, beginnend mit dem Zeichen an der Stelle POS.

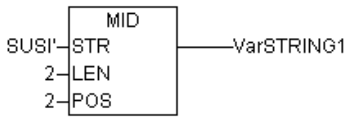
Beispiel in AWL

```
LD   'SUSI'
MID  2,2
ST   VarSTRING1 (* Ergebnis ist 'US' *)
```

Beispiel in ST

```
VarSTRING1 := MID ('SUSI',2,2);
```

Beispiel in FUP



20553AXX

Achtung:
String-Funktionen sind nicht "thread-safe"!

REPLACE

Die Funktion REPLACE der Bibliothek standard.lib ersetzt einen Teilstring eines Strings durch einen anderen String.

Die Eingänge STR1 und STR2 sind vom Typ STRING.

Die Eingänge LEN und POS sind vom Typ INT.

Der Ausgang der Funktion ist vom Typ STRING.

REPLACE (STR1, STR2, L, P) bedeutet: Ersetze L Zeichen aus STR1 durch STR2 beginnend mit dem Zeichen an der Position P.

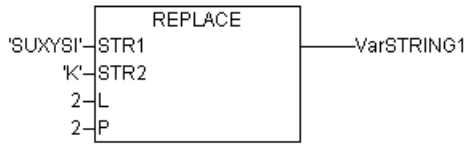
Beispiel in AWL

```
LD      'SUXYSI'
REPLACE 'K',2,2
ST      VarSTRING1 (* Ergebnis ist 'SKYSI' *)
```

Beispiel in ST

```
VarSTRING1 := REPLACE ('SUXYSI','K',2,2);
```

Beispiel in FUP



20557AXX

Achtung:

String-Funktionen sind nicht "thread-safe"!

RIGHT

Die Funktion `RIGHT` der Bibliothek `standard.lib` liefert den rechten Anfangsstring eines Strings.

Der Eingang `STR` ist vom Typ `STRING`.

Der Eingang `SIZE` ist vom Typ `INT`.

Der Ausgang der Funktion ist vom Typ `STRING`.

`RIGHT (STR, SIZE)` bedeutet: Liefere die ersten `SIZE` Zeichen von rechts im String `STR`.

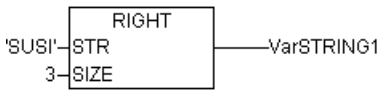
Beispiel in AWL

```
LD    'SUSI'
RIGHT 3
ST    VarSTRING1 (* Ergebnis ist 'USI' *)
```

Beispiel in ST

```
VarSTRING1 := RIGHT ('SUSI',3);
```

Beispiel in FUP



20552AXX

Achtung:

String-Funktionen sind nicht "thread-safe"!

10.2.2 Bistabile Funktionsblöcke

SR

Der Funktionsblock SR der Bibliothek `standard.lib` setzt eine boolesche Variable dominant.

Die Eingänge `SET1` und `RESET` sind vom Typ `BOOL`.

Der Ausgang `Q1` ist vom Typ `BOOL`.

Syntax

```
Q1 = SR(SET1, RESET)
```

Die Syntax ohne die Verwendung von SR:

```
Q1 = (NOT RESET AND Q1) OR SET1
```

Deklarationsbeispiel

```
SRInst : SR;
```

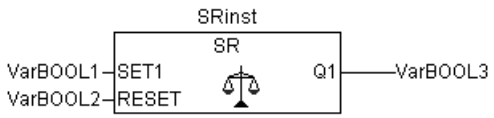
Beispiel in AWL

```
CAL   SRInst(SET1 := VarBOOL1, RESET := VarBOOL2)
LD    SRInst.Q1
ST    VarBOOL3
```

Beispiel in ST

```
SRInst(SET1 := VarBOOL1, RESET := VarBOOL2);
VarBOOL3 := SRInst.Q1;
```

Beispiel in FUP



20559AXX

RS

Der Funktionsblock RS der Bibliothek `standard.lib` setzt eine boolesche Variable dominant zurück.

Die Eingänge SET und RESET1 sind vom Typ BOOL.

Der Ausgang Q1 ist vom Typ BOOL.

Syntax

```
Q1 = RS(SET1, RESET)
```

Die Syntax ohne die Verwendung von RS:

```
Q1 = NOT RESET1 AND (Q1 OR SET)
```

Deklarationsbeispiel

```
RSInst : RS;
```

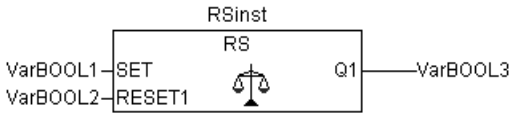
Beispiel in AWL

```
CAL   RSInst(SET := VarBOOL1, RESET1 := VarBOOL2)
LD    RSInst.Q1
ST    VarBOOL3
```

Beispiel in ST

```
RSInst(SET := VarBOOL1, RESET1 := VarBOOL2);
VarBOOL3 := RSInst.Q1;
```

Beispiel in FUP



20560AXX

SEMA

Der Funktionsblock SEMA der Bibliothek standard.lib ist ein Software-Semaphor (= unterbrechbar).

Die interne Variable X ist vom Typ BOOL und ist mit FALSE initialisiert.

Die Eingänge CLAIM und RELEASE sind vom Typ BOOL.

Der Ausgang BUSY ist vom Typ BOOL.

Syntax

```
BUSY = SEMA(CLAIM, RELEASE)
```

Die Syntax ohne die Verwendung von SEMA:

```
BUSY := X;
IF CLAIM THEN
     X := TRUE;
   ELSIF RELEASE THEN
     BUSY := FALSE;
     X := FALSE;
END_IF
```

Wenn der Funktionsblock SEMA aufgerufen wird und der Ausgang BUSY auf TRUE gesetzt ist, wurde der Funktionsblock SEMA bereits vorher aufgerufen wurde und ist belegt. Beim vorherigen Aufruf des Funktionsblocks SEMA war der Eingang CLAIM auf TRUE gesetzt.

Wenn der Ausgang BUSY auf FALSE zurückgesetzt ist, wurde der Funktionsblock SEMA noch nicht aufgerufen, oder er wurde freigegeben durch Setzen des Eingangs RELEASE auf TRUE.

Deklarationsbeispiel

```
SEMAInst : SEMA;
```

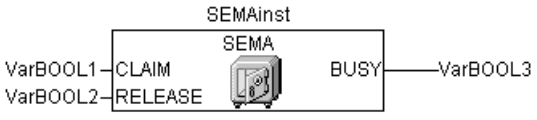
Beispiel in AWL

```
CAL   SEMAInst(CLAIM := VarBOOL1, RELEASE := VarBOOL2)
LD    SEMAInst.BUSY
ST    VarBOOL3
```

Beispiel in ST

```
SEMAInst(CLAIM := VarBOOL1, RELEASE := VarBOOL2);
VarBOOL3 := SEMAInst.BUSY;
```

Beispiel in FUP



20561AXX

10.2.3 Flankenerkennung

R_TRIG

Der Funktionsblock `R_TRIG` der Bibliothek `standard.lib` detektiert eine ansteigende Flanke.

Der Eingang `CLK` ist vom Typ `BOOL`.

Der Ausgang `Q` ist vom Typ `BOOL`.

Syntax

```
FUNCTION_BLOCK R_TRIG
   VAR_INPUT
      CLK : BOOL;
   END_VAR
   VAR_OUTPUT
      Q : BOOL;
   END_VAR
   VAR
      M : BOOL := FALSE;
   END_VAR
   Q := CLK AND NOT M;
   M := CLK;
```

Solange der Eingang `CLK` auf FALSE zurückgesetzt ist, ist der Ausgang `Q` und die Hilfsvariable `M` auf FALSE zurückgesetzt.

Wenn der Eingang `CLK` durch eine steigende Flanke auf TRUE gesetzt wird, wird zuerst der Ausgang `Q` und dann die Hilfsvariable `M` auf TRUE gesetzt.

Bei jedem weiteren Aufruf der Funktionsblockinstanz `R_TRIG` bleibt der Ausgang `Q` auf FALSE, solange bis der Eingang `CLK` durch eine fallende Flanke auf FALSE zurückgesetzt und durch eine steigende Flanke erneut auf TRUE gesetzt wird.

Deklarationsbeispiel

```
R_TRIGInst : R_TRIG ;
```

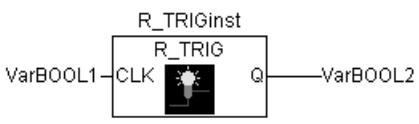
Beispiel in AWL

```
CAL   R_TRIGInst(CLK := VarBOOL1)
LD    R_TRIGInst.Q
ST    VarBOOL2
```

Beispiel in ST

```
R_TRIGInst(CLK := VarBOOL1);
VarBOOL2 := R_TRIGInst.Q;
```

Beispiel in FUP



20562AXX

F_TRIG

Der Funktionsblock F_TRIG der Bibliothek standard.lib detektiert eine fallende Flanke.

Der Eingang CLK ist vom Typ BOOL.

Der Ausgang Q ist vom Typ BOOL.

Syntax

```
FUNCTION_BLOCK F_TRIG
   VAR_INPUT
      CLK : BOOL;
   END_VAR
   VAR_OUTPUT
      Q : BOOL;
   END_VAR
   VAR
      M : BOOL := FALSE;
   END_VAR
   Q := NOT CLK AND NOT M;
   M := NOT CLK;
```

Solange der Eingang CLK auf TRUE gesetzt ist, ist der Ausgang Q und die Hilfsvariable M auf FALSE zurückgesetzt.

Wenn der Eingang CLK durch eine fallende Flanke auf FALSE zurückgesetzt wird, wird zuerst der Ausgang Q und dann die Hilfsvariable M auf TRUE gesetzt.

Bei jedem weiteren Aufruf der Funktionsblockinstanz bleibt der Ausgang Q auf FALSE, bis der Eingang CLK durch eine steigende Flanke auf FALSE und durch eine fallende Flanke erneut auf FALSE zurückgesetzt wird.

Deklarationsbeispiel

```
F_TRIGInst : F_TRIG;
```

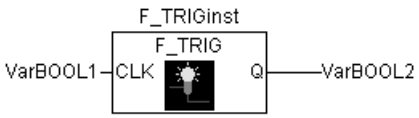
Beispiel in AWL

```
CAL   F_TRIGInst(CLK := VarBOOL1)
LD    F_TRIGInst.Q
ST    VarBOOL2
```

Beispiel in ST

```
F_TRIGInst(CLK := VarBOOL1);
VarBOOL2 := F_TRIGInst.Q;
```

Beispiel in FUP



20563AXX

10.2.4 Zähler

CTU

Der Funktionsblock CTU der Bibliothek `standard.lib` ist ein Aufwärtszähler.

Die Eingänge CU und RESET sind vom Typ BOOL.

Der Eingang PV ist vom Typ INT.

Der Ausgang Q ist vom Typ BOOL.

Der Ausgang CV ist vom Typ INT.

Wenn der Eingang RESET auf TRUE gesetzt ist, wird die Zählvariable am Ausgang CV mit "0" initialisiert.

Wenn der Eingang CU durch eine steigende Flanke von FALSE auf TRUE gesetzt wird, wird die Zählvariable am Ausgang CV um "1" erhöht.

Der Ausgang Q wird auf TRUE gesetzt, wenn die Zählvariable am Ausgang CV größer oder gleich dem am Eingang PV voreingestellten Wert ist.

Deklarationsbeispiel

```
CTUInst : CTU;
```

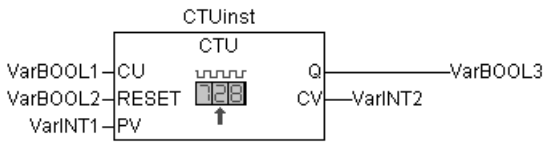
Beispiel in AWL

```
CAL  CTUInst(CU := VarBOOL1, RESET := VarBOOL2, PV := VarINT1)
LD   CTUInst.Q
ST   VarBOOL3
LD   CTUInst.CV
ST   VarINT2
```

Beispiel in ST

```
CTUInst(CU := VarBOOL1, RESET := VarBOOL2, PV := VarINT1);
VarBOOL3 := CTUInst.Q;
VarINT2 := CTUInst.CV;
```

Beispiel in FUP



20564AXX

CTD

Der Funktionsblock CTD der Bibliothek standard.lib ist ein Abwärtszähler.

Die Eingänge CD und LOAD sind vom Typ BOOL.

Der Eingang PV ist vom Typ INT.

Der Ausgang Q ist vom Typ BOOL.

Der Ausgang CV ist vom Typ INT.

Wenn der Eingang LOAD auf TRUE gesetzt ist, wird die Zählvariable am Ausgang CV mit dem am Eingang PV voreingestellten Wert initialisiert.

Wenn der Eingang CD durch eine steigende Flanke von FALSE auf TRUE gesetzt wird, wird die Zählvariable am Ausgang CV um "1" verringert. Die Zählvariable am Ausgang CV wird nur verringert, solange sie größer "0" ist, also kein Unterlauf verursacht wird.

Der Ausgang Q wird auf TRUE gesetzt, wenn die Zählvariable am Ausgang CV gleich "0" ist.

Deklarationsbeispiel

```
CTDInst : CTD;
```

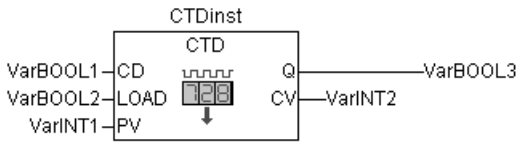
Beispiel in AWL

```
CAL   CTDInst(CD := VarBOOL1, LOAD := VarBOOL2, PV := VarINT1)
LD    CTDInst.Q
ST    VarBOOL3
LD    CTDInst.CV
ST    VarINT2
```

Beispiel in ST

```
CTDInst(CD := VarBOOL1, LOAD := VarBOOL2, PV := VarINT1);
VarBOOL3 := CTDInst.Q;
VarINT2 := CTDInst.CV;
```

Beispiel in FUP



20565AXX

CTUD

Der Funktionsblock CTUD der Bibliothek standard.lib ist ein Auf- und Abwärtszähler.

Die Eingänge CU, CD, RESET und LOAD sind vom Typ BOOL.

Der Eingang PV ist vom Typ INT.

Die Ausgänge QU und QD sind vom Typ BOOL.

Der Ausgang CV ist vom Typ INT.

Wenn der Eingang RESET auf TRUE gesetzt wird, wird die Zählvariable am Ausgang CV mit "0" initialisiert.

Wenn der Eingang LOAD auf TRUE gesetzt ist, wird die Zählvariable am Ausgang CV mit dem am Eingang PV voreingestellten Wert initialisiert.

Wenn der Eingang CU durch eine steigende Flanke von FALSE auf TRUE gesetzt wird, wird die Zählvariable am Ausgang CV um "1" erhöht.

Wenn der Eingang CD durch eine steigende Flanke von FALSE auf

TRUE gesetzt wird, dann wird die Zählvariable am Ausgang CV um "1" verringert. Die Zählvariable am Ausgang CV wird nur verringert, solange sie größer "0" ist, also kein Unterlauf verursacht wird.

Der Ausgang QU wird auf TRUE gesetzt, wenn die Zählvariable am Ausgang CV größer oder gleich dem am Eingang PV voreingestellten Wert ist.

Der Ausgang QD wird auf TRUE gesetzt, wenn die Zählvariable am Ausgang CV gleich "0" ist.

Deklarationsbeispiel

```
CTUDInst : CUTD;
```

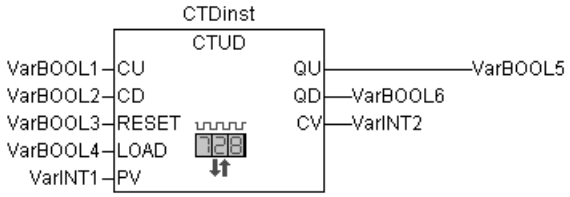
Beispiel in AWL

```
CAL   CTUDInst(CU := VarBOOL2, RESET := VarBOOL3, LOAD := VarBOOL4,
               PV := VarINT1)
LD    CTUDInst.Q
ST    VarBOOL5
LD    CTUDInst.QD
ST    VarBOOL5
LD    CTUDInst.CV
ST    VarINT2
```

Beispiel in ST

```
CTUDInst(CU := VarBOOL1, CU := VarBOOL2, Reset := VarBOOL3,
         LOAD := VarBOOL4, PV := VarINT1);
VarBOOL5 := CTUDInst.QU;
VarBOOL6 := CTUDInst.QD;
VarINT2 := CTUDInst.CV;
```

Beispiel in FUP



20566AXX

10.2.5 Timer

TP

Der Funktionsblock TP der Bibliothek `standard.lib` ist ein Pulsgeber.

Der Eingang IN ist vom Typ BOOL. Der Eingang PT ist vom Typ TIME.

Der Ausgang Q ist vom Typ BOOL. Der Ausgang ET ist vom Typ TIME.

TP (IN, PT, Q, ET) bedeutet:

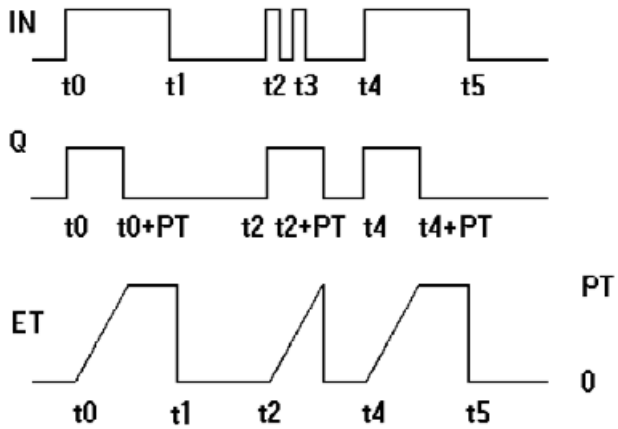
Wenn der Eingang IN auf FALSE gesetzt ist, wird der Ausgang Q ebenfalls auf FALSE und der Ausgang ET auf "0" gesetzt.

Wenn der Eingang IN auf TRUE gesetzt ist, wird am Ausgang ET die Zeit in Millisekunden hochgezählt, bis der Wert gleich dem am Eingang PV voreingestellten Wert ist.

Der Ausgang Q ist auf TRUE gesetzt, wenn der Eingang IN auf TRUE gesetzt ist und die Zeit am Ausgang ET kleiner oder gleich dem am Eingang PV voreingestellten Wert ist. Andernfalls ist der Ausgang Q auf FALSE gesetzt.

Der Ausgang Q liefert also für die am Eingang PV voreingestellte Zeit ein Signal.

Grafische Darstellung des zeitlichen Ablaufs von TP



20567AXX

Deklarationsbeispiel

```
TPInst : TP;
```

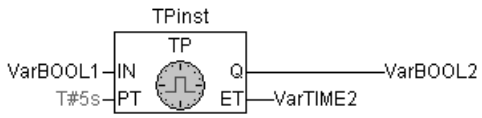
Beispiel in AWL

```
CAL   TPInst(IN := VarBOOL1, PT := T#5s)
LD    TPInst.Q
ST    VarBOOL2
```

Beispiel in ST

```
TPInst(IN := VarBOOL1, PT := T#5s);
VarBOOL2 := TPInst.Q;
```

Beispiel in FUP



20568AXX

TON

Der Funktionsblock TON der Bibliothek standard.lib realisiert eine Einschaltverzögerung.

Der Eingang IN ist vom Typ BOOL. Der Eingang PT ist vom Typ TIME.

Der Ausgang Q ist vom Typ BOOL. Der Ausgang ET ist vom Typ TIME.

TON (IN, PT, Q, ET) bedeutet:

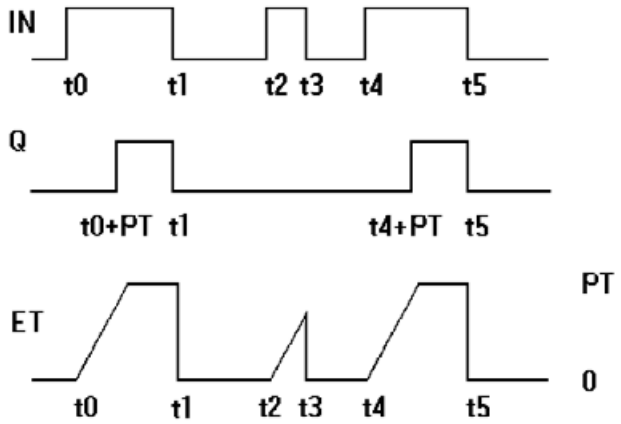
Wenn der Eingang IN auf FALSE gesetzt ist, wird der Ausgang Q ebenfalls auf FALSE und der Ausgang ET auf "0" gesetzt.

Wenn der Eingang IN auf TRUE gesetzt ist, wird am Ausgang ET die Zeit in Millisekunden hochgezählt, bis der Wert gleich dem am Eingang PV voreingestellten Wert ist.

Der Ausgang Q ist auf TRUE gesetzt, wenn der Eingang IN auf TRUE gesetzt ist und die Zeit am Ausgang ET gleich dem am Eingang PV voreingestellten Wert ist. Andernfalls ist der Ausgang Q auf FALSE gesetzt.

Der Ausgang Q liefert also eine steigende Flanke, sobald die am Eingang PV voreingestellte Zeit abgelaufen ist.

Grafische Darstellung des zeitlichen Ablaufs von TON



20569AXX

Deklarationsbeispiel

```
TONInst : TON;
```

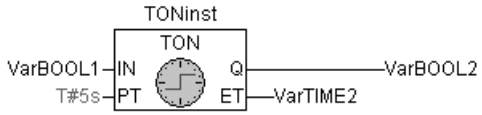
Beispiel in AWL

```
CAL  TONInst(IN := VarBOOL1, PT := T#5s)
LD   TONInst.Q
ST   VarBOOL2
```

Beispiel in ST

```
TONInst(IN := VarBOOL1, PT := T#5s);
VarBOOL2 := TONInst.Q;
```

Beispiel in FUP



20570AXX

TOF

Der Funktionsblock TOF der Bibliothek standard.lib realisiert eine Ausschaltverzögerung.

Der Eingang IN ist vom Typ BOOL. Der Eingang PT ist vom Typ TIME.

Der Ausgang Q ist vom Typ BOOL. Der Ausgang ET ist vom Typ TIME.

TOF (IN, PT, Q, ET) bedeutet:

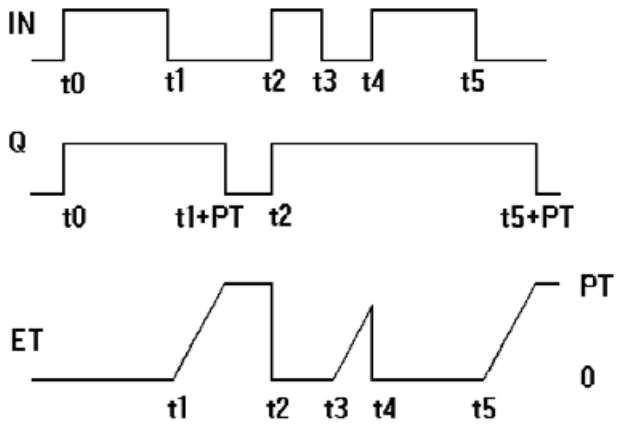
Wenn der Eingang IN auf TRUE gesetzt ist, wird der Ausgang Q ebenfalls auf TRUE und der Ausgang ET auf "0" gesetzt.

Wenn der Eingang IN auf TRUE gesetzt ist, wird am Ausgang ET die Zeit in Millisekunden hochgezählt, bis der Wert gleich dem am Eingang PV voreingestellten Wert ist.

Der Ausgang Q wird auf FALSE gesetzt, wenn der Eingang IN auf FALSE gesetzt ist und die Zeit am Ausgang ET gleich dem am Eingang PV voreingestellten Wert ist. Andernfalls ist der Ausgang Q auf TRUE gesetzt.

Der Ausgang Q liefert also eine fallende Flanke sobald die am Eingang PV voreingestellte Zeit abgelaufen ist.

Grafische Darstellung des zeitlichen Ablaufs von TOF



20571AXX

Deklarationsbeispiel

```
TOFInst : TOF;
```

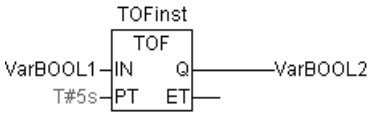
Beispiel in AWL

```
CAL   TOFInst(IN := VarBOOL1, PT := T#5s)
LD    TOFInst.Q
ST    VarBOOL2
```

Beispiel in ST

```
TOFInst(IN := VarBOOL1, PT := T#5s);
VarBOOL2 := TOFInst.Q;
```

Beispiel in FUP



20572AXX

RTC

Der Funktionsblock RTC der Bibliothek standard.lib liefert die Uhrzeit und das Datum fortlaufend ab einem vorgegebenen Startzeitpunkt.

Der Eingang EN ist vom Typ BOOL.

Der Eingang PDT ist vom Typ DATE_AND_TIME.

Der Ausgang Q ist vom Typ BOOL.

Der Ausgang CDT ist vom Typ DATE_AND_TIME.

RTC (EN, PDT, Q, CDT) bedeutet:

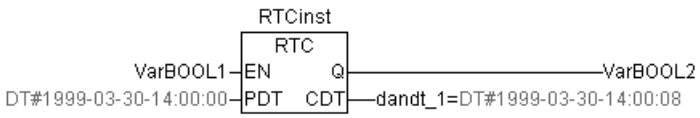
Wenn der Eingang EN auf FALSE zurückgesetzt ist, wird der Ausgang Q ebenfalls auf FALSE zurückgesetzt und der Ausgang CDT liefert den Wert DT#1970-01-01-00-00:00:00.

Wenn der Eingang EN auf TRUE gesetzt ist, wird die am Eingang PDT anliegende Zeit übernommen. Die Sekunden werden hochgezählt und die ktuelle Zeit am Ausgang CDT ausgegeben.

Sobald der Eingang EN wieder auf FALSE gesetzt wird, liefert der Ausgang CDT wieder den Wert DT#1970-01-01-00-00:00:00.

Achtung:

Die am Eingang PDT anliegende Zeit wird nur am Ausgang CDT ausgegeben, wenn am Eingang EN eine steigende Flanke auftritt!



20573AXX

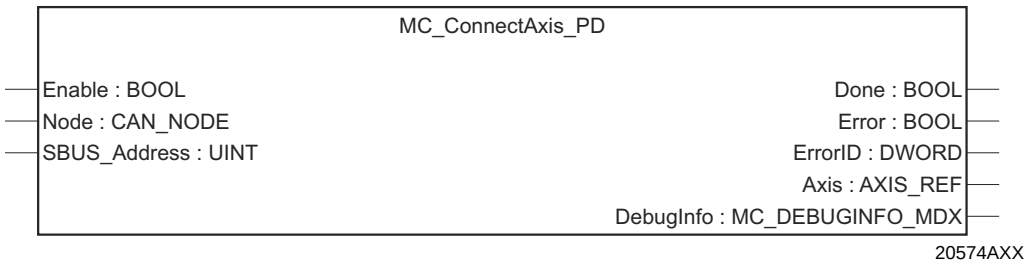
10.3 Bibliothek MPLCProcessdata.lib

Die Bibliothek `MPLCProcessdata.lib` realisiert eine Prozessdaten-Schnittstelle zu den Antrieben. Wie bei anderen externen Steuerungen, erfolgt die Kommunikation zwischen der Steuerung MOVIPLC® und den Antriebsumrichtern über den Prozessdaten- und Parameterkanal.

Die Bibliothek `MPLCProcessdata.lib` ermöglicht Folgendes:

- Ansteuerung aller SEW-Umrichter über Prozessdaten
- Lesen bzw. Schreiben von Umrichterparametern
- Prozessdatenprofile; SEW-Applikationsmodule oder auch kundenspezifische IPOS®-Programme können weiterhin verwendet werden.

10.3.1 Funktionsbaustein MC_ConnectAxis_PD



Beschreibung

Der Funktionsbaustein `MC_ConnectAxis_PD` dient zur Übertragung von bis zu vier Prozessdaten an einen SEW Umrichter:

- MOVITRAC®07x
- MOVIDRIVE®
- MOVIAXIS®

In Verbindung mit dem Antriebsumrichter MOVIDRIVE® kann ein freies IPOS®-Programm wie von einer beliebigen externen Steuerung angesprochen werden.

Die Prozessdaten werden unabhängig vom aufrufenden Taskzyklus alle fünf Millisekunden übertragen.

Anwendung

Der Funktionsbaustein `MC_ConnectAxis_PD` ist anwendbar auf alle Achsen mit Systembus.

Voraussetzungen

In der Steuerungskonfiguration wurde ein SEW-Processdata-Modul an einem CAN-Strang angehängt. Um die Prozessdaten im Anwenderprogramm zu lesen oder zu schreiben, können Sie hier individuelle Namen eingeben.

Der Funktionsbaustein `MC_ConnectAxis_PD` muss im Anwenderprogramm zyklisch aufgerufen werden.

Hinweis:

Die Daten werden in den Prozessdatenpuffer des Antriebsumrichters MOVIDRIVE® geschrieben. Die Bedeutung wird am Antriebsumrichter MOVIDRIVE® festgelegt.

Eingänge

Der Funktionsbaustein `MC_ConnectAxis_PD` verfügt über folgende Eingänge:

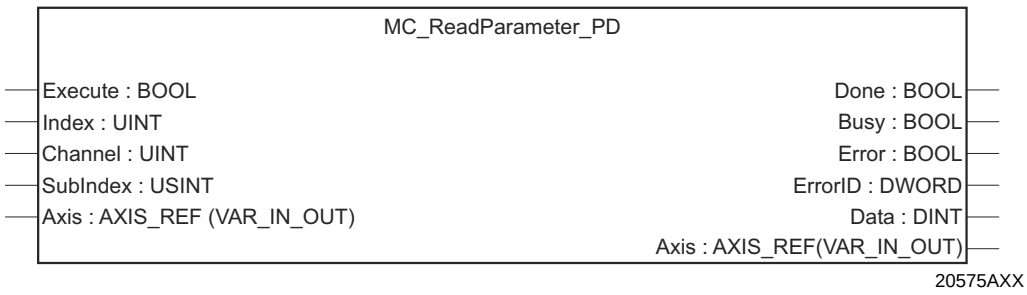
Eingang	Typ	Bedeutung
Enable	BOOL	Dieser Eingang dient zur Aktivierung des Funktionsbausteins.
Node	CAN_ NODE	Dieser Eingang dient zur Einstellung des CAN-Bus-Knotens, an dem der Antriebsumrichter angeschlossen ist. • SBUS_NODE_1: CAN 1 (Stecker X33 bei DHP11B) • SBUS_NODE_2: CAN 2 (Stecker X32 bei DHP11B)
SBUS_Address	UINT	Dieser Eingang dient zur Angabe der Systembus-Adresse des angeschlossenen Antriebsumrichters.

Ausgänge

Der Funktionsaustein `MC_ConnectAxis_PD` verfügt über folgende Ausgänge:

Ausgang	Typ	Bedeutung
Done	BOOL	Dieser Ausgang zeigt an, ob die Initialisierung des Funktionsbausteins beendet ist. • TRUE: Der Funktionsbaustein `MC_ConnectAxis_PD` hat die Initialisierung beendet und die Kommunikationsverbindung zum Antriebsumrichter aufgebaut. • FALSE: Die Initialisierung ist noch nicht beendet.
Error	BOOL	Dieser Ausgang zeigt an, ob ein Fehler aufgetreten ist. • TRUE: Während der Aktion des Funktionsbausteins ist ein Fehler aufgetreten. • FALSE: Es ist kein Fehler aufgetreten.
ErrorID	DWORD	Dieser Ausgang zeigt den Fehlercode des aufgetretenen Fehlers an (→ siehe Kapitel Error Identifier).
Axis	AXIS_REF	Dieser Ausgang zeigt die Achsreferenz an. Das Ausgangssignal muss an alle Funktionsbausteine übergeben werden, die sich auf diese physikalische Motorachse beziehen.
DebugInfo	MC_ DEBUG INFO_MDX	Dieser Ausgang dient zur Lokalisierung eventueller Fehler im Operation-Mode Debug (Einstellung in der Steuerungskonfiguration bei den Modulparametern des Antriebsumrichters). • `ParameterChannelIdleTime`: Dieser Wert zeigt die Verfügbarkeit des Parameterkanals in % an. Pro Motorachse wird in jedem Steuerungs-Zyklus maximal ein Parametertelegramm bearbeitet. 0% → Ein Parametertelegramm in jedem MOVI-PLC®-Zyklus 100% → Kein Parametertelegramm • `PDODebugInfo`: `Counter` und `TimeStamp` von PDE und dann von PDA. Hier sind nur vierWörter gültig. • `NumberOfReceiveErrors`: Anzahl der Fehler beim Empfangen der Daten. • `NumberOfTransmitErrors`: Anzahl der Fehler beim Senden der Daten.

10.3.2 Funktionsbaustein MC_ReadParameter_PD



Beschreibung Der Funktionsbaustein `MC_ReadParameter_PD` dient zum Einlesen der Parameter (Indizes) von einem SEW-Umrichter zur Steuerung MOVI-PLC®.

Anwendung Der Funktionsbaustein `MC_ReadParameter_PD` ist anwendbar auf alle Motorachsen mit Systembus.

Voraussetzungen Der Funktionsbaustein `MC_ConnectAxis_PD` ist aufgerufen.

Eingänge Der Funktionsbaustein `MC_ReadParameter_PD` verfügt über folgende Eingänge:

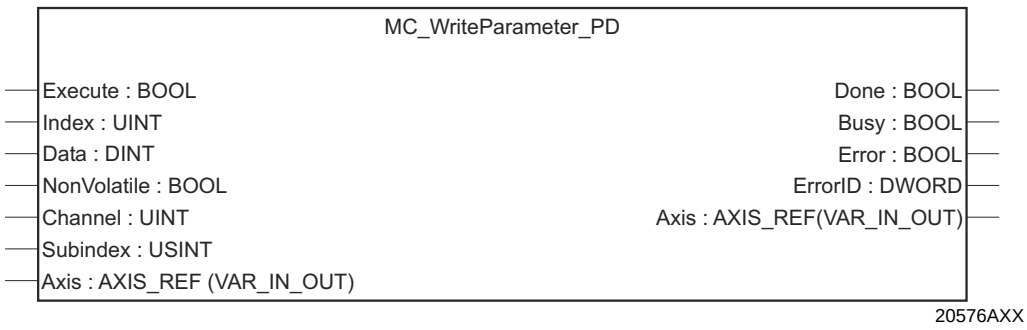
Eingang	Typ	Bedeutung
Execute	BOOL	Dieser Eingang dient zum Starten des Lesevorgangs des Parameters. Wenn an diesem Eingang eine steigende Flanke auftritt, startet der Funktionsbaustein die Übertragung des Parameters vom Umrichter zur Steuerung MOVI-PLC®.
Index	UINT	Dieser Eingang legt fest, welcher Parameter vom Umrichter zur Steuerung MOVI-PLC® übertragen wird. Die Shell der Software MOVITOOLS®-MotionStudio zeigt die Indexnummer des Parameters an, wenn Sie den Cursor auf das entsprechende Parameterfeld verschieben und die Tasten <Strg> + <F1> drücken.
Channel	UINT	Dieser Eingang legt fest, über welchen Parameterkanal der Parameter übertragen wird. • 1: Übertragung über den Parameterkanal 1. • 2: Übertragung über den Parameterkanal 2.
SubIndex	USINT	Dieser Eingang legt den SubIndex des Index (für MOVIAXIS®) fest.
Axis	AXIS_REF	Dieser Eingang legt fest, an welcher Motorachse die Aktionen des Funktionsbausteins ausgeführt werden.

Ausgänge

Der Funktionsbaustein `MC_ReadParameter_PD` verfügt über folgende Ausgänge:

Ausgang	Typ	Bedeutung
Done	BOOL	Dieser Ausgang zeigt an, ob der Parameter erfolgreich übertragen wurde. • TRUE: Der Wert des Parameters am Ausgang `Data` ist gültig. • FALSE: Der Parameter wurde nicht übertragen.
Busy	BOOL	Dieser Ausgang zeigt an, ob der Parameter momentan übertragen wird. • TRUE: Der Parameter wird momentan übertragen. • FALSE: Der Parameter wird momentan nicht übertragen.
Error	BOOL	Dieser Ausgang zeigt an, ob im Funktionsbaustein ein Fehler aufgetreten ist. • TRUE: Während der Übertragung des Parameters ist ein Fehler aufgetreten. • FALSE: Es ist kein Fehler aufgetreten.
ErrorID	DWORD	Dieser Ausgang zeigt den Fehlercode des aufgetretenen Fehlers an (→ siehe Kapitel Error Identifier).
Data	DINT	Dieser Ausgang enthält den übertragenen Wert des Parameters.

10.3.3 Funktionsbaustein MC_WriteParameter_PD



20576AXX

Beschreibung

Der Funktionsbaustein `MC_WriteParameter_PD` dient zum Übertragen der Parameter (Indizes) von der Steuerung MOVI-PLC® zu einem SEW-Umrichter.

Anwendung

Der Funktionsbaustein `MC_WriteParameter_PD` ist anwendbar auf alle Achsen mit Systembus.

Voraussetzung

Der Funktionsbaustein `MC_ConnectAxis_PD` ist aufgerufen.

Eingänge

Der Baustein `MC_WriteParameter_PD` verfügt über folgende Eingänge:

Eingang	Typ	Bedeutung
Execute	BOOL	Dieser Eingang dient zum Starten der Übertragung des Parameters. Wenn an diesem Eingang eine steigende Flanke auftritt, startet der Funktionsbaustein die Übertragung des Parameters von der Steuerung MOVI-PLC® zum Umrichter.
Index	UINT	Dieser Eingang legt fest, welcher Parameter von der Steuerung MOVI-PLC® zum Umtriebsumrichter übertragen wird. Die Shell der Software MOVITOOLS®-MotionStudio zeigt die Indexnummer des Parameters an, wenn Sie den Cursor auf das entsprechende Parameterfeld verschieben und die Tasten <Strg>+<F1> drücken.
Data	DINT	Dieser Eingang enthält den Wert des übertragenen Parameters.
NonVolatile	BOOL	Dieser Eingang legt fest, ob der Parameter flüchtig übertragen werden soll. • TRUE: Der Parameter wird **nicht** flüchtig an den Umrichter übertragen. Der Wert des Parameters bleibt nach dem Aus- und wieder Einschalten des Umrichters erhalten. • FALSE: Der Parameter wird flüchtig an den Umrichter übertragen. Der Parameter wird beim Aus- und wieder Einschalten des Umrichters auf seinen ursprünglichen Wert gesetzt.
Channel	UINT	Dieser Eingang legt fest, über welchen Parameterkanal der Parameter übertragen wird. • 1: Übertragung über den Parameterkanal 1. • 2: Übertragung über den Parameterkanal 2. Der Parameterkanal 2 wird momentan nur von MOVIAXIS® unterstützt.
SubIndex	USINT	Dieser Einhang legt den SubIndex des Index (für MOVIAXIS®) fest.
Axis	AXIS_REF	Dieser Eingang legt fest, an welcher Motorachse die Aktionen des Funktionsbausteins ausgeführt werden.

Ausgänge

Der Baustein `MC_WriteParameter_PD` verfügt über folgende Ausgänge:

Ausgang	Typ	Bedeutung
Done	BOOL	Dieser Ausgang zeigt an, ob der Parameter erfolgreich übertragen wurde. • TRUE: Der Parameter wurde erfolgreich von der Steuerung MOVI-PLC® zum Umrichter übertragen. • FALSE: Der Parameter wurde nicht übertragen.
Busy	BOOL	Dieser Ausgang zeigt an, ob der Parameter momentan übertragen wird. • TRUE: Der Parameter wird momentan übertragen. • FALSE: Der Parameter wird momentan nicht übertragen.
Error	BOOL	Dieser Ausgang zeigt an, ob im Funktionsbaustein ein Fehler aufgetreten ist. • TRUE: Während der Übertragung des Parameters ist ein Fehler aufgetreten. • FALSE: Es ist kein Fehler aufgetreten.
ErrorID	DWORD	Dieser Ausgang zeigt den Fehlercode des aufgetretenen Fehlers an.

11 Anhang

11.1 Tastaturbedienung

11.1.1 Hinweis

Sie können den PLC-Editor ausschießlich über die Tastatur bedienen. Es gibt Tastaturbefehle, die Menübefehlen zugeordnet und im PLC-Editor rechts neben dem Menübefehl dargestellt sind.

Dazu gibt es noch weitere Tastaturbefehle:

<F6>:

- Wechsel zwischen Deklarationsteil und Anweisungsteil eines geöffneten Bausteins
- Wechsel zwischen Navigationsfenster und Listeneditor im Parametermanager

<Alt>+<F6>

- Wechsel von einem geöffneten Objekt zum Object Organizer; beim erneuten Ausführen dieser Tastenkombination Wechsel in das Meldungsfenster, falls dieses geöffnet ist.
- Wechsel vom Object Organizer zum Dialog [Suchen], falls dieser geöffnet ist. Beim erneuten Ausführen diesem Tastaturbefehl Wechsel zum Objekt.

<Tabulator>

Mit der <Tabulator>-Taste springen Sie innerhalb eines Dialogs zum nächsten Eingabefeld bzw. zur nächsten Schaltfläche.

<Umschalt>+<Tabulator>

Mit diesem Tastaturbefehle springen Sie innerhalb eines Dialogs zum vorhergehenden Eingabefeld bzw. zur vorgehenden Schaltfläche.

<Pfeiltasten>

Mit den <Pfeiltasten> bewegen Sie sich innerhalb des Object Organizer und des Bibliotheksverwalters durch die Registerkarten und die Objekte.

Alle anderen Aktionen können über die Menübefehle oder über die Kurzformen, die sich hinter den Menübefehlen befinden, ausgelöst werden.

Mit **<Umschalt>+<F10>** (oder alternativ mit der rechten Maustaste) erhalten Sie das Kontextmenü mit den am häufigsten verwendeten Befehlen für ein markiertes Objekt oder für den aktiven Editor.

11.1.2 Tastaturbefehl

In diesem Kapitel finden Sie eine Übersicht aller Tastenkombinationen und Funktionstasten.

Aktion	Tastaturbefehl
Allgemeine Bedienung	
Wechsel zwischen Deklarationsteil und Anweisungsteil eines Bausteins	<F6>
Wechsel zwischen Object Organizer, Objekt und Meldungsfenster	<Alt>+<F6>
Kontextmenü	<Umschalt>+<F10>
Wechsel zum nächsten geöffneten Editorfenster	<Strg>+<F6>
Wechsel zum vorherigen geöffneten Editorfenster	<Strg>+<Umschalt>+<F6>
Kurzformmodus für Deklarationen	<Strg>+<Eingabe>
Wechsel von einer Meldung im Meldungsfenster zu der dazugehörigen Stelle im Editor	<Eingabe>
Auf- und Zuklappen mehrstufiger Variablen	<Eingabe>
Auf- und Zuklappen von Ordnern	<Eingabe>
Registerkartenwechsel im Object Organizer und Bibliotheksverwalter	<Pfeiltasten>
Weiterspringen in Dialogen	<Tabulator>
Kontextsensitive Hilfe	<F1>
Allgemeine Menübefehle	
[Datei] / [Speichern]	<Strg>+<S>
[Datei] / [Drucken]	<Strg>+<P>
[Datei] / [Beenden]	<Alt>+<F4>
[Projekt] / [Alles Überprüfen]	<Strg>+<F11>
[Projekt] / [Übersetzen]	<Umschalt>+<F11>
[Projekt] / [Alles Übersetzen]	<F11>
[Projekt] / [Objekt löschen]	<Entf>
[Projekt] / [Objekt einfügen]	<Einfg>
[Projekt] / [Objekt umbenennen]	<Leertaste>
[Projekt] / [Objekt bearbeiten]	<Eingabe>
[Bearbeiten] / [Rückgängig]	<Strg>+<Z>
[Bearbeiten] / [Wiederherstellen]	<Strg>+<Y>
[Bearbeiten] / [Ausschneiden]	<Strg>+<X> oder <Umschalt>+<Entf>
[Bearbeiten] / [Kopieren]	<Strg>+<C>
[Bearbeiten] / [Einfügen]	<Strg>+<V>
[Bearbeiten] / [Löschen]	<Entf>
[Bearbeiten] / [Weitersuchen]	<F3>
[Bearbeiten] / [Eingabehilfe]	<F2>
[Bearbeiten] / [Variablendeklaration]	<Umschalt><F2>
[Bearbeiten] / [Nächster Fehler]	<F4>
[Bearbeiten] / [Vorheriger Fehler]	<Umschalt>+<F4>
[Online] / [Einloggen]	<Alt><F8>
[Online] / [Ausloggen]	<Strg><F8>
[Online] / [Start]	<F5>
[Online] / [Stop]	<Umschalt>+<F8>
[Online] / [Breakpoint an/aus]	<F9>
[Online] / [Einzelschritt über]	<F10>
[Online] / [Einzelschritt in]	<F8>

Aktion	Tastaturbefehl
[Online] / [Einzelzyklus]	<Strg>+<F5>
[Online] / [Werte schreiben]	<Strg>+<F7>
[Online] / [Werte forcen]	<F7>
[Online] / [Forcen aufheben]	<Umschalt>+<F7>
[Online] / [Schreiben/Forcen-Dialog]	<Strg>+<Umschalt>+<F7>
[Online] / [Meldungen]	<Umschalt>+<Esc>
Befehle des FUP-Editors	
[Einfügen] / [Netzwerk (danach)]	<Strg>+<T>
[Einfügen] / [Eingang]	<Strg>+<U>
[Einfügen] / [Baustein]	<Strg>+<B>
[Einfügen] / [Zuweisung]	<Strg>+<A>
[Einfügen] / [Sprung]	<Strg>+<L>
[Einfügen] / [Return]	<Strg>+<R>
[Einfügen] / [Operator]	<Strg>+<O>
[Einfügen] / [Funktion]	<Strg>+<F>
[Einfügen] / [Funktionsblock]	<Strg>+<B>
[Extras] / [Negation]	<Strg>+<N>
[Extras] / [Zoom]	<Alt>+<Eingabe>
Befehle des CFC-Editors	
[Einfügen] / [Eingang]	<Strg>+<E>
[Einfügen] / [Ausgang]	<Strg>+<A>
[Einfügen] / [Sprung]	<Strg>+<G>
[Einfügen] / [Label]	<Strg>+<L>
[Einfügen] / [Return]	<Strg>+<R>
[Einfügen] / [Kommentar]	<Strg>+<K>
[Einfügen] / [Bausteineingang]	<Strg>+<U>
[Extras] / [Negation]	<Strg>+<N>
[Extras] / [Set/Reset]	<Strg>+<T>
[Extras] / [Verbindung]	<Strg>+<M>
[Extras] / [EN/ENO]	<Strg>+<E>
[Extras] / [Zoom]	<Alt>+<Eingabe>
Befehle des KOP-Editors	
[Einfügen] / [Netzwerk (danach)]	<Umschalt>+<T>
[Einfügen] / [Kontakt]	<Strg>+<K>
[Einfügen] / [Paralleler Kontakt]	<Strg>+<R>
[Einfügen] / [Funktionsblock]	<Strg>+<B>
[Einfügen] / [Spule]	<Strg>+<L>
[Extras] / [Darunter Einfügen]	<Strg>+<U>
[Extras] / [Negation]	<Strg>+<N>
[Extras] / [Zoom]	<Alt>+<Eingabe>
Befehle des AS-Editors	
[Einfügen] / [Schritt-Transition (davor)]	<Strg>+<T>
[Einfügen] / [Schritt-Transition (danach)]	<Strg>+<E>

Aktion	Tastaturbefehl
[Einfügen] / [Alternativzweig (rechts)]	<Strg>+<A>
[Einfügen] / [Parallelzweig (rechts)]	<Strg>+<L>
[Einfügen] / [Sprung (AS)]	<Strg>+<U>
[Extras] / [Zoom Aktion/Transition]	<Alt>+<Eingabe>
Wechsel aus AS-Übersicht zurück in Editor	<Eingabe>
Bedienung der Steuerungs- bzw. Taskkonfiguration	
Auf- und Zuklappen von Organisationselementen	<Eingabe>

11.2 Schlüsselwörter im PLC-Editor

Die folgenden Zeichenfolgen sind im PLC-Editor als Schlüsselwörter reserviert. Schlwüsselwörter können nicht als Bezeichner für Variablen oder Bausteine verwendet werden.

`ACTION` (nur im Export-Format verwendet)

`ADD`

`END_ACTION` (nur im Export-Format verwendet)

`END_CASE`

`END_FUNCTION` (nur im Export-Format verwendet)

`END_FUNCTION_BLOCK` (nur im Export-Format verwendet)

`END_IF`

`END_PROGRAM` (nur im Export-Format verwendet)

`END_REPEAT`

`END_ONLY`

`READ_WRITE`

`REAL`

`VAR_CONFIG`

`WSTRING` (IEC-Datentyp, im PLC-Editor nicht unterstützt)

`XOR`

11.3 PLC-Editor-Dateien

Die folgenden Dateien können aus dem PLC-Editor erzeugt werden.

Datei	Beispiel	Beschreibung	Format	Ablageort (Default)
*.alm	alarmalog0.alm	Alarmdatei		benutzerdefiniertes Verzeichnis oder Download-Verzeichnis der Steuerung
*.asd	project01.asd	Automatisch erzeugte, temporäre Sicherungsdatei; erzeugt vor Übersetzungslauf Einstellung über Menübefehl [Projekt] / [Optionen] / [Laden & Speichern] / [Automatisch sichern vor Übersetzen]	binär	Projektverzeichnis
*.asm	code386.asm	Assembler-Listing des erzeugten Codes	Text	Übersetzungsverzeichnis (compile)
*.bak	project01.bak	Automatisch erzeugte, permanente Sicherungskopie des Projekts Einstellung über Menübefehl [Projekt] / [Optionen] / [Laden & Speichern]	binär	Projektverzeichnis
*.bin	project01.bin	Bootprojekt; Benennung vom Zielsystem abhängig	binär	Zielsystem, wenn online erzeugt; Projektverzeichnis, wenn offline erzeugt
*.bpl	project01.bpl	Debug-Datei (Breakpoint-Informationen)	Text	Übersetzungsverzeichnis (compile)
*.ci	project01<Zahl>.ci	Informationen über den letzten Übersetzungslaufs – "Incremental Compile" möglich; nur beim Speichern erzeugt; Zahl = kodierte Target-ID	binär	Projektverzeichnis
*.dfr	default.dfr, project01.dfr	Rahmendatei (Einstellungen Dokumentation)	binär	bei PLCEditor.exe
*.exp	project01.exp, PLC_PRG.exp	Exportdatei; mit Menübefehl [Projekt] / [Exportieren] erzeugt	Exportformat (Text)	Projektverzeichnis
*.h86	project01.h86 (Output) bzw. standard.h86 (Lib)	Compiler-Output oder -Input für externe Bibliothek; für Intel	Intel-h86-Datei	Übersetzungsverzeichnis (compile) bzw. Bibliotheksverzeichnis
*.hex	project01.hex (Output) bzw. standard.hex (Lib)	Compiler-Output oder -Input für externe Bibliothek; für Intel und Motorola	Intel- oder Motorola-hex-Datei	Übersetzungsverzeichnis (compile) bzw. Bibliotheksverzeichnis
*.ini	PLCEditor.ini	Datei für diverse PLC-Editor-Einstellungen	Text	bei PLCEditor.exe
*.lst	project01.lst	Assembler-Listing des erzeugten Codes	Text	Übersetzungsverzeichnis (compile)
*.log	projectXY.log	Logbuchdatei	binär	Projektverzeichnis
*.map	project01.map	Informationen zur Speicheraufteilung und "variable locations"	Text	Übersetzungsverzeichnis (compile)
*.mon	project01_tr1.mon	Traceaufzeichnung	XML	bei PLCEditor.exe
*.pro	project01.pro	Projektdatei	binär	Projektverzeichnis
*.ri	project01<Zahl>.ri	Informationen über den letzten Download; wichtig für "Online Change"; bei jedem Download erzeugt Zahl = kodierte Target-ID	binär	Projektverzeichnis
*.sdb	project01.sdb	Symboldatei	binär	Projektverzeichnis
*.st	PLC_PRG.st	Debug-Datei, implizierter ST-Code	Text	Übersetzungsverzeichnis (compile)
*.sym	project01.sym	Symboldatei	Text	Projektverzeichnis
*.sym_xml	project01.sym_xml	Symboldatei	XML	Projektverzeichnis
*.tcf	project01_tr1.tcf	Traceaufzeichnung	binär	bei PLCEditor.exe
*.tlt	project01.tlt	Übersetzungsdatei; über Menübefehl [Projekt] / [In andere Sprache übersetzen]	Text	

Datei	Beispiel	Beschreibung	Format	Ablageort (Default)
*.trc	project01_tr1.trc	Traceaufzeichnung	binär	bei PLCEditor.exe
*.txt	project01.txt	Übersetzungsdatei; über Menübefehl [Projekt] / [In andere Sprache übersetzen] erzeugt	Text	
*.wtc	projX_watch1.wtc	Watchliste	Text	benutzerdefiniertes Verzeichnis
*.zip	projectXY.zip	Archivdatei mit projektzugehörigen Dateien; mit Menübefehl [Datei] / [Archiv speichern/versenden] erzeugt		benutzerdefiniertes Verzeichnis

11.4 Warnungen und Übersetzungsfehler

11.4.1 Hinweis

Beim Kompilieren des Projekts werden Meldungen zu aufgetretenen Übersetzungsfehlern bzw. Warnungen im Meldungsfenster ausgegeben. Mit der <F4>-Taste springen Sie zur nächsten Meldung. Gleichzeitig wird ein Fenster mit der dazugehörigen Programmstelle geöffnet. Den Fehlermeldungen und Warnungen sind im Meldungsfenster ID-Nummern vorangestellt.

Wenn eine Meldung im Meldungsfenster markiert ist, können Sie durch Drücken der <F1>-Taste das dazugehörige Hilfefenster öffnen.

11.4.2 Warnungen

1100 **Unbekannte Funktion <Name> in Bibliothek.**

Sie verwenden eine externe Bibliothek. Überprüfen Sie, ob alle Funktionen, die in der `*.hex`-Datei angegeben sind, auch in der `*.lib`-Datei definiert sind.

1101 **Nicht aufgelöstes Symbol <Symbol>.**

Der Codegenerator erwartet einen Baustein mit dem Namen <Symbol>. Dieser ist im Projekt nicht definiert. Definieren Sie eine Funktion/ein Programm mit dem entsprechenden Namen.

1102 **Ungültige Schnittstelle für Symbol <Symbol>.**

Der Codegenerator erwartet einen der folgenden Programmteile:

- eine Funktion mit dem Namen <`Symbol`> und genau einem skalaren Eingang
- ein Programm mit dem Namen <`Symbol`> und keinem Ein- oder Ausgang

1103 **Die Konstante <Name> an Code-Adresse <%04X %04X> liegt über einer 16K-Seitengrenze!**

Eine Stringkonstante liegt über der Begrenzung von 16 kB für die Seitengröße. Das System kann dies nicht handhaben. Abhängig vom Laufzeitsystem besteht eventuell die Möglichkeit, dies über einen Eintrag in der Targetdatei zu umgehen. Bitte wenden Sie sich diesbezüglich an Ihren Steuerungshersteller.

1200 **Task <Name>, Aufruf von <Name> Accessvariablen in der Parameterliste werden nicht aktualisiert**.

Variablen, die nur bei einem Funktionsbaustein-Aufruf in der Taskkonfiguration verwendet werden, werden nicht in die Querverweisliste eingetragen.

1300 **Die Datei <Name> wurde nicht gefunden**.

Die Datei, auf die das globale Variablenobjekt verweist, existiert nicht. Prüfen Sie den Pfad.

1301 **Analyse-Bibliothek wird nicht gefunden. Code für Analyse wird nicht erzeugt.**

Sie verwenden die Funktion `Analyze`, die Bibliothek `analyzation.lib` fehlt jedoch. Fügen Sie die Bibliothek im Bibliotheksverwalter ein.

1302 **Neue extern referenzierte Funktionen eingefügt. Online Change ist damit nicht mehr möglich!**

Sie haben seit dem letzten Download eine Bibliothek eingebunden, die Funktionen enthält, die im Laufzeitsystem noch nicht referenziert sind. Deshalb ist ein Download des gesamten Projekts nötig.

1400 **Unbekannte Compilerdirektive <Name> wird ignoriert!**

Dieses Pragma wird vom Compiler nicht unterstützt. Informationen zu unterstützten Direktiven finden Sie auf Seite 212 (→ Kap. 5.3, Pragma-Anweisungen).

1401 **Die Struktur <Name> enthält keine Elemente.**

Die Struktur enthält keine Elemente. Variablen dieses Typs belegen jedoch 1 Byte im Speicher.

1410 **RETAIN und PERSISTENT haben in Funktionen keinen Effekt.**

Die in einer Funktion als remanent deklarierten Variablen werden nicht im `RETAIN`-Bereich gespeichert, sondern wie normale lokale Variablen behandelt.

1411 **Variable <name> in der Variablenkonfiguration wird in keiner Task aktualisiert**.

Die oberste Instanz der Variable wird in keinem Task über einen Aufruf referenziert und somit auch nicht vom Prozessabbild kopiert.

Beispiel

Variablenkonfiguration:

```
VAR_CONFIG
  plc_prg.aprg.ainst.in AT %IB0 : INT;
END_VAR
```

Baustein `PLC_PRG`:

```
index := INDEXOF(aprg);
```

Das Program `aprg` wird zwar referenziert, aber nicht aufgerufen. Deshalb wird `plc_prg.aprg.ainst.in` nie den wirklichen Wert von `%IB0` enthalten.

1412 **Unerwartetes Token <Name> in Pragma {Pragmaname}**

Sie verwenden einen Pragma-Namen, der nicht korrekt ist, bzw. ein Pragma, das an dieser Stelle nicht anwendbar ist. Nehmen Sie Korrekturen gegebenenfalls anhand der Beschreibungen ab Seite 212 (→ Kap. 5.3, Pragma-Anweisungen) vor.

1413 **<Name> ist kein gültiger Schlüssel für Liste <Name>. Der Schlüssel wird ignoriert.**

Sie geben im Pragma eine nicht vorhandene Parameterliste an. Überprüfen Sie den Listennamen bzw. sehen Sie im Parameter-Manager nach, welche Listen verfügbar sind.

1414 **Zu viele Komponentendefinitionen in Pragma <pragmaanweisung>**

Das Pragma enthält mehr Definitionen (in eckigen Klammern) als Element im Array, im Funktionsblock oder in der Struktur enthalten sind.

1500 **Diese Expression enthält keine Zuweisung. Es wird kein Code generiert.**

Das Ergebnis dieses Ausdrucks wird nicht verwendet. Somit wird für den gesamten Ausdruck kein Code generiert.

1501 **String Konstante wird als VAR_IN_OUT übergeben: <Name> darf nicht überschrieben werden!**

Die Konstante darf im Rumpf des Bausteins nicht beschrieben werden, da dort keine Größenprüfung möglich ist.

1502 **Variable <Name> hat den gleichen Namen wie ein Baustein. Der Baustein wird nicht aufgerufen!**

Sie verwenden eine Variable, die den gleichen Namen wie ein Baustein hat.

Beispiel

```
PROGRAM A
...
VAR_GLOBAL
  A:INT;
END_VAR
...
A; (* Es wird nicht der Baustein A aufgerufen, sondern die Variable A
geladen. *)
```

1503 **Der Baustein hat keine Ausgänge, Verknüpfung wird mit TRUE fortgesetzt.**

Sie verknüpfen den Ausgangs-Pin eines Bausteins, der keine Ausgänge hat, in FUP oder KOP weiter. Die Verknüpfung bekommt automatisch den Wert TRUE zugewiesen.

1504 **Anweisung wird möglicherweise nicht ausgeführt, abhängig vom logischen Ausdruck**.

Unter Umständen werden nicht alle Zweige des logischen Ausdrucks ausgeführt.

Beispiel

```
IF a AND funct(TRUE) THEN ....
```

Wenn a den Wert FALSE hat, wird `funct` nicht mehr aufgerufen.

1505 **Seiteneffekt in <Name>! Zweig wird möglicherweise nicht gerechnet**.

Der erste Eingang des Bausteins ist FALSE. Deshalb wird der Seitenzweig, der am zweiten Eingang einmündet, eventuell nicht mehr berechnet.

1506 **Variable <Name> hat den gleichen Namen wie eine lokale Aktion. Die Aktion wird nicht aufgerufen!**

Benennen Sie die Variable oder die Aktion um, damit diese unterschiedliche Namen haben.

1507 **Instanz <Name> heißt wie eine Funktion. Die Instanz wird nicht aufgerufen.**

Sie rufen in der Programmiersprache ST eine Instanz auf, die den gleichen Namen hat wie eine Funktion. Es wird die Funktion aufgerufen! Benennen Sie die Instanz oder die Funktion um, damit diese unterschiedliche Namen haben.

1550 **Der mehrmalige Aufruf desselben Bausteins <Name> kann zu unerwünschten gegenseitigen Beeinflussungen führen.**

Überprüfen Sie, ob die mehrfache Verwendung des Bausteins wirklich nötig ist. Durch den mehrfachen Aufruf in einem Zyklus können unerwünschte Werteüberschreibungen auftreten.

1600 **Offene DB unklar (Generierter Code kann fehlerhaft sein).**

Aus dem original Siemens-Programm geht nicht hervor, welcher Datenbaustein geöffnet ist.

1700 **Eingang nicht verbunden.**

Sie verwenden im CFC eine Eingangsbox, die nicht weiterverbunden ist. Es wird dafür kein Code erzeugt.

1750 **Schritt <Name>: die minimale Zeit ist größer als die maximale Zeit!**

Öffnen Sie den Dialog [Schrittattribute] zum angegebenen Schritt und korrigieren Sie die Zeitangaben.

1751 **Vorsicht bei Verwendung der Variable <Name>. Diese Variable wird von implizit erzeugtem Code verwendet und beeinflusst den Ablauf der Schrittkette!**

Benennen Sie die Variable sicherheitshalber um, sodass sie einen eindeutigen Bezeichner erhält und unerwünschte Effekte vermieden werden können.

1800 **<Name> (Element #<Elementnummer>): Ungültiger Watchausdruck <Name>**

Das Visualisierungselement enthält einen Ausdruck, für den das Monitoring nicht angewendet werden kann. Prüfen Sie Variablennamen und Platzhalterersetzungen.

1801 **Eingabe auf Ausdruck nicht möglich.**

Sie verwenden in der Konfiguration des Visualisierungsobjekts einen zusammengesetzten Ausdruck als Ziel einer Eingabeaktion. Ersetzen Sie diesen durch eine einzelne Variable.

1802 **<Visualisierungsobjekt> (Elementnummer): Bitmap <Name> wurde nicht gefunden.**

Stellen Sie sicher, dass die externe Bitmap-Datei entsprechend dem in der Visualisierung definierten Verknüpfungspfad vorliegt.

1803 **<name> (<number>): Die Aktion Drucken wird für die Target- und Web-Visualisierung nicht unterstützt.**

Ein in der Visualisierung konfigurierter Alarm ist mit der Aktion [Drucken] verknüpft. Dies wird in Web- oder Target-Visualisierung nicht berücksichtigt.

1804 **<Name> (<Nummer>): Der Zeichensatz <Name> wird vom Zielsystem nicht unterstützt.**

Sie verwenden in der Visualisierung einen Font, der vom Zielsystem nicht unterstützt wird. Die vom aktuellen Zielsystem unterstützten Zeichensätze werden in den Zielsystemeinstellungen, Kategorie [Visualisierung] angezeigt.

1805 **<Name> (<Nummer>): Trenddatenaufzeichnung innerhalb der Steuerung sollte aktiviert werden.**

Sie verwenden ein Visualisierungselement zur Trenddatenaufzeichnung. Dies wird jedoch auf der Steuerung nicht berücksichtigt, da die Option [Trenddatenaufzeichnung] in den Zielsystemeinstellungen, Kategorie [Visualisierung] nicht aktiviert ist.

1806 **<Name> (<Nummer>): Die Zielsystemeinstellung [Alarmbehandlung innerhalb der Steuerung] sollte aktiviert werden.**

Sie verwenden in der Visualisierung ein Element zur Alarmvisualisierung. Dies wird jedoch auf der Steuerung nicht berücksichtigt, da die Option [Alarmbehandlung innerhalb der Steuerung] in den Zielsystemeinstellungen, Kategorie [Visualisierung] nicht aktiviert ist.

1850 **Eingangsvariable an %IB<Zahl> wird in Task <Name> benutzt, aber in anderem Task aktualisiert.**

Prüfen Sie, welche Tasks die angegebene Variable verwenden und ob die augenblickliche Programmierung zu unerwünschten Effekten führt. Die Aktualisierung des Variablenwerts erfolgt in der Regel durch die Task mit der höheren Priorität.

1851 **Ausgangsvariable an %QB<Zahl> wird in Task <Name> benutzt, aber in anderem Task aktualisiert.**

Prüfen Sie, welche Tasks die angegebene Variable verwenden und ob die augenblickliche Programmierung zu unerwünschten Effekten führt. Die Aktualisierung des Variablenwerts erfolgt in der Regel durch die Task mit der höheren Priorität.

1852 **CanOpenMaster wird in Ereignistask <Name> u. U. nicht zyklisch aufgerufen! Modulparameter UpdateTask setzen!**

Momentan wird der CanOpen-Master durch die genannte Ereignistask aufgerufen. Wenn ein zyklischer Aufruf gewünscht ist, definieren Sie eine entsprechende Task über Parameter UpdateTask in der Steuerungskonfiguration im Dialog [Modulparameter].

1853 **Ein PDO (Index: <Zahl>) wird in Ereignistask <Name> u. U. nicht zyklisch aktualisiert.**

Momentan wird das genannte PDO durch die genannte Ereignistask aufgerufen. Wenn jedoch ein zyklischer Aufruf gewünscht ist, weisen Sie dem PDO eine entsprechende Task zu, indem Sie IO-Referenzen in diese Task verschieben.

1900 **Die POU <Name> (Einsprungfunktion) steht in der Bibliothek nicht zur Verfügung.**

Der Einsprungsbaustein (z. B. PLC_PRG) steht beim Verwenden der Bibliothek nicht zur Verfügung.

1901 **Access Variablen und Konfigurationsvariablen werden nicht in einer Bibliothek abgespeichert!**

Accessvariablen und Konfigurationsvariablen werden nicht in der Bibliothek gespeichert.

1902 **<Name>: Bibliothek für den aktuellen Maschinentyp nicht geeignet!**

Die `*.obj`-Datei der Bibliothek wurde für einen anderen Maschinentyp erzeugt.

1903 **<Name>: Ungültige Bibliothek**

Die Bibliotheken-Datei entspricht nicht dem benötigten Dateiformat des Zielsystems.

1904 **Die Konstante <Name> verschattet eine gleichnamige Konstante in der Bibliothek**

Sie haben im Projekt eine Konstante deklariert, die den gleichen Namen hat wie eine Konstante in einer eingebundenen Bibliothek. Die Bibliotheksvariable wird überschrieben!

1970 **Parametermanager: Liste <Name>, Spalte <Name>, Wert <Name> konnte nicht importiert werden!**

Überprüfen Sie die Import-Datei `*.prm` auf Einträge, die nicht zur aktuellen Konfiguration (Standardwerte bzw. XML-Beschreibungsdatei) des Parameter-Managers passen.

1980 **Globale Netzwerk-Variablen <Name>, <Name>: Gleichzeitiges Lesen und Schreiben kann zu Datenverlust führen!**

Sie haben bei der Konfiguration der Netzwerkvariablenliste sowohl die Option [Lesen] als auch die Option [Schreiben] aktiviert . Bedenken Sie, dass dies eventuell zu Datenverlusten bei der Kommunikation führen kann.

1990 **Kein VAR_CONFIG für <Name>**

Für diese Variable liegt keine Addresskonfiguration in der Variablenkonfiguration (`VAR_CONFIG`) vor. Öffnen Sie im Object Organizer im Register [Ressourcen] das Fenster [`VAR_CONFIG`] und fügen Sie die Adresskonfiguration ein (Menübefehl [Einfügen] / [Alle Instanzpfade]).

11.4.3 Übersetzungsfehler

3100 **Programm zu groß. Maximale Größe: <Anzahl> Byte (<Anzahl> kB)**

Die maximale Programmgröße ist überschritten. Verkleinern Sie das Programm.

3101 **Datenbereich zu groß. Maximale Größe: <Anzahl> Byte (<Anzahl>K)**

Der Datenspeicher ist aufgebraucht. Verringern Sie den Datenbedarf Ihrer Applikation.

3110 **Fehler in Bibliotheks-Datei <Name>.**

Die `*.hex`-Datei entspricht nicht dem Intel hex-Format.

3111 **Bibliothek <Name> ist zu groß. Maximale Größe: 64K**

Die `*.hex`-Datei überschreitet die maximal mögliche Größe.

3112 **Nicht verschiebbare Anweisung in Bibliothek.**

Die `*.hex`-Datei enthält eine nicht relozierbare (verschiebbare) Anweisung. Der Bibliotheks-Code kann nicht gelinkt werden.

3113 **Bibliotheks-Code überschreibt Funktionstabellen.**

Die Bereiche für Code und Informationstabellen überlappen sich.

3114 **Bibliothek verwendet mehr als ein Segment.**

Die in der `*.hex`-Datei enthaltenen Tabellen und der Code verwenden mehr als ein Segment.

3115 **Konstante kann nicht an VAR_IN_OUT zugewiesen werden. Inkompatible Datentypen.**

Das interne Zeigerformat für Stringkonstanten kann nicht in das interne Zeigerformat von `VAR_IN_OUT` konvertiert werden, weil die Daten als "near" und die Stringkonstanten als "huge" oder "far" definiert sind. Wenn möglich, verändern Sie diese Zielsystemeinstellungen.

3116 **Funktionstabellen überschreiben Bibliotheks-Code oder Segmentgrenze.**

Code 166x: Die externe Bibliothek kann so nicht verwendet werden. Die Zielsystemeinstellungen für die externen Bibliotheken müssen angepasst werden bzw. die Bibliothek mit anderen Einstellungen neu erzeugt werden.

3117 **<Bausteinname> (<Zeilennummer>): Ausdruck zu komplex. Keine Register mehr verfügbar.**

Der genannte Ausdruck ist so komplex, dass die verfügbaren Register nicht ausreichen. Versuchen Sie, den Ausdruck durch Verwendung von Zwischenvariablen zu vereinfachen.

3120 **Aktuelles Code-Segment ist größer als 64K.**

Der soeben generierte System-Code ist größer als 64 kB. Eventuell wird zuviel Initialisierungs-Code benötigt.

3121 **Baustein zu groß.**

Ein Baustein darf die Größe von 64 kB nicht überschreiten.

3122 **Initialisierung zu groß. Maximale Größe: 64K**

Der Initialisierungscode für einen Funktionsbaustein oder eine Struktur darf 64 kB nicht überschreiten.

3123 **Datensegment zu groß: Segment <Nummer>, Größe <Name des Datenbereichs> Bytes (Maximum <Name des Datenbereichs> Bytes)**

Wenden Sie sich an Ihren Hardware-Hersteller.

3130 **Anwendungs-Stack zu klein: <Anzahl> DWORD benötigt, <Anzahl> DWORD verfügbar.**

Die Schachtelungstiefe der Bausteinaufrufe ist zu groß. Vergrößern Sie die Stackgröße in den Zielsystemeinstellungen oder übersetzen Sie das Programm ohne die Projektübersetzungsoption [Debug].

3131 **Benutzer-Stack zu klein: <Anzahl> WORD benötigt, <Anzahl> WORD verfügbar.**

Wenden Sie sich an Ihren Steuerungshersteller.

3132 **System-Stack zu klein: <Anzahl> WORD benötigt, <Anzahl> WORD verfügbar.**

Wenden Sie sich an Ihren Steuerungshersteller.

3150 **Parameter <Zahl> der Funktion <Name>: Das Ergebnis einer IEC-Funktion kann nicht als Stringparameter einer C-Funktion übergeben werden.**

Verwenden Sie eine Zwischenvariable, auf die das Ergebnis der IEC-Funktion gelegt wird.

3160 **Kann Bibliotheks-Datei <Name> nicht öffnen.**

Die für eine Bibliothek benötigte Datei <Name> kann nicht gefunden werden.

3161 **Bibliothek <Name> enthält kein Codesegment**

Eine `*.obj`-Datei einer Bibliothek muss mindestens eine C-Funktion enthalten. Fügen Sie in die `*.obj`-Datei eine Dummy-Funktion ein, die nicht in der `*.lib`-Datei definiert ist.

3162 **Kann Referenz in Bibliothek <Name> nicht auflösen (Symbol <Name>, Class <Name>, Type <Name>)**

Die `*.obj`-Datei enthält eine nicht auflösbare Referenz auf ein anderes Symbol. Prüfen Sie die Einstellungen des C-Compilers.

3163 **Unbekannter Referenztyp in Bibliothek <Name> (Symbol <Name>, Class <Name>, Type <Name>)**

Die `*.obj`-Datei enthält einen vom Codegenerator nicht auflösbaren Referenztypen. Prüfen Sie die Einstellungen des C-Compilers.

3200 **<Name> (<Zahl>): Logischer Ausdruck zu komplex.**

Der temporäre Speicher des Zielsystems reicht für die Größe des Ausdrucks nicht aus. Teilen Sie den Ausdruck in mehrere Teilausdrücke mit Zuweisungen auf Zwischenvariablen auf.

3201 **<Name> (<Netzwerk>): Ein Netzwerk darf maximal 512 Bytes Code ergeben.**

Interne Sprünge können nicht aufgelöst werden. Aktivieren Sie die Option [16 bit Sprungoffsets verwenden] in den [68k target settings].

3202 **Stacküberlauf bei geschachtelten String/Array/Struktur-Funktionsaufrufen.**

Sie benutzen einen geschachtelten Funktionsaufruf der Form `CONCAT(x, f(i))`. Dies kann zu Datenverlust führen. Teilen Sie den Aufruf in zwei Ausdrücke auf.

3203 **Zuweisung zu komplex (zu viele benötigte Adressregister).**

Teilen Sie die Zuweisung in mehrere auf.

3204 **Ein Sprung ist über 32k Bytes lang.**

Sprungdistanzen dürfen nicht größer als 32767 Byte sein.

3205 **Interner Fehler: Zu viele constant strings.**

In einem Baustein dürfen maximal 3000 Stringkonstanten verwendet werden.

3206 **Funktionsblock zu groß.**

Ein Funktionsblock darf maximal 32767 Byte Code ergeben.

3207 **Array Optimierung**

Die Optimierung der Array-Zugriffe schlug fehl, weil innerhalb der Indexberechnung eine Funktion aufgerufen wurde.

3208 **Umwandlung ist nicht implementiert.**

Sie verwenden eine Konvertierungsfunktion, die für den aktuellen Codegenerator nicht implementiert ist.

3209 **Operator nicht implementiert.**

Sie verwenden einen Operator, der für diesen Datentyp beim aktuellen Codegenerator nicht implementiert ist: `MIN(string1,string2)`.

3210 **Funktion <Name> nicht gefunden.**

Sie rufen eine Funktion auf, die im Projekt nicht vorhanden ist.

3211 **Stringvariable zu oft verwendet.**

Eine Variable vom Typ `String` darf beim 68K-Codegenerator in einem Ausdruck nur zehnmal verwendet werden.

3212 **Falsche Bibliotheksreihenfolge bei Baustein <Bausteinname>.**

Die Reihenfolge der Bibliotheken für den Baustein <Bausteinname> im Bibliotheksmanager stimmt nicht mit der in der Datei cslib.hex überein. Korrigieren Sie die Reihenfolge entsprechend. (Nur für 68K-Zielsysteme, wenn die Prüfung der Reihenfolge in der Target-Datei aktiviert ist.)

3250 **Real wird auf 8 Bit Controller nicht unterstützt.**

Das Zielsystem wird derzeit nicht unterstützt.

3251 **"date of day"-Typen werden auf 8 Bit Controllern nicht unterstützt.**

Das Zielsystem wird derzeit nicht unterstützt.

3252 **Stackgröße übersteigt <Zahl> Bytes.**

Das Zielsystem wird derzeit nicht unterstützt.

3253 **Hexfile nicht gefunden: <Name>.**

Das Zielsystem wird derzeit nicht unterstützt.

3254 **Aufruf einer externen Bibliotheksfunktion konnte nicht aufgelöst werden.**

Das Zielsystem wird derzeit nicht unterstützt.

3255 **Pointer werden auf 8 Bit Controllern nicht unterstützt.**

Um das Programm auf dem 8-Bit-Rechner in Betrieb nehmen zu können, müssen Sie die Verwendung von Pointern umgehen.

3400 **Fehler beim Importieren der Access Variablen.**

Die *.exp-Datei enthält einen fehlerhaften Accessvariablen-Abschnitt.

3401 **Fehler beim Importieren der Variablenkonfiguration.**

Die *.exp-Datei enthält einen fehlerhaften Variablenkonfigurations-Abschnitt.

3402 **Fehler beim Importieren der globalen Variablen.**

Die *.exp-Datei enthält einen fehlerhaften Abschnitt der globalen Variablen.

3403 **<Name> konnte nicht importiert werden.**

Der Abschnitt in der *.exp-Datei für das angegebene Objekt ist fehlerhaft.

3404 **Fehler beim Importieren der Taskkonfiguration.**

Der Abschnitt in der *.exp-Datei für die Taskkonfiguration ist fehlerhaft.

3405 **Fehler beim Importieren der Steuerungskonfiguration.**

Der Abschnitt in der *.exp-Datei für die Steuerungskonfiguration ist fehlerhaft.

3406 **Zwei Schritte mit dem Namen <Name>. Der zweite Schritt wurde nicht importiert.**

Der in der *.exp-Datei enthaltene Abschnitt des AS-Bausteins enthält zwei Schritte mit gleichem Namen. Benennen Sie einen der Schritte in der Export-Datei um.

3407 **Eingangsschritt <Name> nicht gefunden.**

In der *.exp-Datei fehlt der genannte Schritt.

3408 **Nachfolgeschritt <Name> nicht gefunden.**

In der *.exp-Datei fehlt der genannte Schritt.

3409 **Keine nachfolgende Transition für Schritt <Name>.**

In der *.exp-Datei fehlt eine Transition, die den genannten Schritt als Eingangsschritt benötigt.

3410 **Kein nachfolgender Schritt für Transition <Name>.**

In der *.exp-Datei fehlt ein Schritt, der die genannte Transition benötigt.

3411 **Schritt <Name> nicht erreichbar von Init-Step.**

In der *.exp-Datei fehlt die Verknüpfung zwischen dem genannten Schritt und dem Init-Step.

3412 **Makro <Name> konnte nicht importiert werden.**

Überprüfen Sie die Exportdatei.

3413 **Fehler beim Importieren der Kurvenscheiben.**

Sie haben eine Exportdatei *.exp importiert, die fehlerhafte Informationen zu einer Kurvenscheibe enthält. Überprüfen Sie die Exportdatei.

3414 **Fehler beim Importieren der CNC-Programmliste.**

Sie haben eine Exportdatei *.exp importiert, die fehlerhafte Informationen zu einem CNC-Programm enthält. Überprüfen Sie die Exportdatei.

3415 **Fehler beim Importieren der Alarmkonfiguration.**

Sie haben eine Exportdatei *.exp importiert, die fehlerhafte Informationen zur Alarmkonfiguration enthält. Überprüfen Sie die Exportdatei.

3450 **PDO <Name>: <Modulname> <Konfigurationsdialogname>-<PDO Name> COB-ID fehlt!**

Klicken Sie im Konfigurationsdialog [Eigenschaften] des Moduls <Modulname> auf die Schaltfläche [Eigenschaften] und geben Sie für die PDO <Name> eine COB-ID ein.

3451 **Fehler beim Laden: EDS-Datei <Name> konnte nicht gefunden werden, wird aber in der Konfiguration verwendet!**

Die für die CAN-Konfiguration benötigte Gerätedatei liegt eventuell nicht im richtigen Verzeichnis. Prüfen Sie dies anhand des Verzeichniseintrags für Konfigurationsdateien im Menübefehl [Projekt] / [Optionen] / [Verzeichnisse].

3452 **Das Modul <Name> konnte nicht erstellt werden!**

Die Gerätedatei für Modul <Name> passt nicht zur vorliegenden Konfiguration. Eventuell wurde die Gerätedatei seit Erstellung der Konfiguration verändert oder sie ist korrupt.

3453 **Der Kanal <Name> konnte nicht erstellt werden!**

Die Gerätedatei für Kanal <Name> passt nicht zur vorliegenden Konfiguration. Eventuell wurde die Gerätedatei seit Erstellung der Konfiguration verändert oder sie ist korrupt.

3454 **Die Adresse <Name> verweist auf einen belegten Speicherbereich!**

Sie haben die Option [Adressüberschneidungen prüfen] im Dialog [Einstellungen] der [Steuerungskonfiguration] aktiviert und es wurde eine Überschneidung festgestellt. Beachten Sie, dass die Grundlage der Bereichsprüfung die Größe ist, die sich aufgrund des Datentyps der Module ergibt und nicht der Wert im Eintrag [size] in der Konfigurationsdatei!

3455 **Fehler beim Laden: GSD-Datei <Name> konnte nicht gefunden werden, wird aber in der Konfiguration verwendet!**

Die für die Profibus-Konfiguration nötige Gerätedatei liegt eventuell nicht im richtigen Verzeichnis vor. Siehe hierzu Eintrag für Konfigurationsdateien im Menübefehl [Projekt] / [Optionen] / [Verzeichnisse].

3456 **Das Profibus-Gerät <Name> konnte nicht erstellt werden!**

Die Gerätedatei für das Gerät <Name> passt nicht mehr zur vorliegenden Konfiguration. Eventuell wurde die Gerätedatei seit Erstellung der Konfiguration verändert oder sie ist korrupt.

3457 **Modulbeschreibung fehlerhaft: <Name>!**

Prüfen Sie die zum Modul gehörige Gerätedatei.

3458 **Die Steuerungskonfiguration konnte nicht erstellt werden! Überprüfen Sie die Konfigurationsdateien.**

Prüfen Sie, ob alle nötigen Konfigurations- und Gerätedateien im eingestellten Konfigurationsdateien-Verzeichnis (Menübefehl [Projekt] / [Optionen] / [Verzeichnisse]) vorliegen.

3459 **Die eingestellte Baudrate wird nicht unterstützt!**

Ändern Sie die Einstellung im CAN-Parameter-Dialog. Beachten Sie die Angaben zur Baudrate in der GSD-Datei.

3460 **3S_CanDrv.lib hat die falsche Version.**

Stellen Sie sicher, dass Sie die aktuelle Version der `3S_CanDrv.lib` im Projekt eingebunden haben.

3461 **3S_CanOpenMaster.lib hat die falsche Version.**

Stellen Sie sicher, dass Sie die aktuelle Version der `3S_CanOpenMaster.lib` im Projekt eingebunden haben.

3462 **3S_CanOpenDevice.lib hat die falsche Version.**

Stellen Sie sicher, dass Sie die aktuelle Version der `3S_CanOpenDevice.lib` im Projekt eingebunden haben.

3463 **3S_CanOpenManager.lib hat die falsche Version.**

Stellen Sie sicher, dass Sie die aktuelle Version der `3S_CanOpenManager.lib` im Projekt eingebunden haben.

3464 **3S_CanNetVar.lib hat die falsche Version.**

Stellen Sie sicher, dass Sie die aktuelle Version der `3S_CanNetVar.lib` im Projekt eingebunden haben.

3465 **CanDevice: Subindices müssen sequentiell nummeriert werden.**

In Parameterlisten, die im CanDevice verwendet werden, müssen die Subindizes der Einträge fortlaufend und lückenlos nummeriert sein. Prüfen Sie die entsprechende Liste im Parameter-Manager.

3466 **CAN-Netzwerkvariablen: Kein CAN-Controller in der Steuerungskonfiguration gefunden.**

Sie haben Netzwerkvariablen für ein CAN-Netzwerk konfiguriert (Ressourcen, Globale Variablen), in der Steuerungskonfiguration des Projekts ist jedoch kein CAN-Controller verfügbar.

3468 **CanDevice: Name der UpdateTask nicht in Taskkonfiguration vorhanden.**

Der Parameter UpdateTask (für Aufruf des CanDevice), der im Dialog [Grundeinstellungen] für das CanDevice in der Steuerungskonfiguration definiert ist, muss in der Taskkonfiguration des Projekts konfiguriert sein.

3469 **Der CanOpenMaster-Baustein kann nicht aufgerufen werden. Task bitte manuell zuordnen.**

Weisen Sie dem CanOpenMaster-Baustein über den Parameter UpdateTask im Dialog [Modulparameter] in der Steuerungskonfiguration eine Task zu, über die der Baustein aufgerufen werden soll.

3470 **Ungültiger Name in Parameter UpdateTask.**

Öffnen Sie den Dialog [Modulparameter] in der Steuerungskonfiguration für den CAN-Master. Prüfen Sie den für den Parameter UpdateTask angegebenen Tasknamen. Die angegebene Task muss im Projekt verfügbar sein. Eventuell reicht es, eine andere Task einzustellen, eventuell müssen Sie aber auch die Wertevorgabe durch die Gerätedatei prüfen.

3500 **Kein VAR_CONFIG für <Name>.**

Erstellen Sie für die genannte Variable in der globalen Variablenliste, die die Variablenkonfiguration enthält, eine Variablendeklaration.

3501 **Keine Adresse in VAR_CONFIG für <Name>.**

Fügen Sie für die genannte Variable in der globalen Variablenliste, die die Variablenkonfiguration enthält, eine Adresse ein.

3502 **Falscher Datentyp von <Name> in VAR_CONFIG.**

Die genannte Variable ist in der globalen Variablenliste, die die Variablenkonfiguration enthält, mit einem anderen Datentypen deklariert als im Funktionsbaustein.

3503 **Falscher Adresstyp von <Name> in VAR_CONFIG.**

Die genannte Variable ist in der globalen Variablenliste, die die Variablenkonfiguration enthält, mit einem anderen Adresstyp deklariert als im Funktionsbaustein.

3504 **Initialwerte für VAR_CONFIG Variablen werden nicht unterstützt.**

Eine Variable der Variablenkonfiguration ist mit Adresse und Initialwert deklariert. Ein Initialwert kann jedoch nur bei Eingangsvariablen ohne Adresszuweisung definiert werden.

3505 **<Name> ist kein gültiger Instanzpfad.**

In der Variablenkonfiguration wurde eine Variable angegeben, die nicht existiert.

3506 **Zugriffspfad erwartet.**

In der globalen Variablenliste für die Accessvariablen fehlt für eine Variable ein korrekter Zugriffspfad in der Form:

```
<Bezeichner>:'<Zugriffspfad>':<Typ> <Zugriffsart>
```

3507 **Keine Adressangabe für VAR_ACCESS erlaubt.**

In der globalen Variablenliste für die Accessvariablen ist für eine Variable eine nicht erlaubte Adresszuweisung vorhanden.

Gültige Definition: `<Bezeichner>:'<Zugriffspfad>':<Typ> <Zugriffsart>`

3550 **Der Taskname <Name> wurde doppelt verwendet.**

Sie haben zwei Tasks mit demselben Namen definiert. Benennen Sie eine davon um.

3551 **Die Task <Name> muss mindestens einen Programmaufruf enthalten.**

Fügen Sie einen Programmaufruf ein oder löschen Sie die Task.

3552 **Ereignis-Variable <Name> in Task <Name> nicht definiert.**

Sie haben in der Konfiguration der genannten Task eine Ereignis-Variable verwendet, die im Projekt nicht global deklariert ist. Verwenden Sie eine andere Variable bzw. deklarieren Sie die angegebene Variable global.

3553 **Ereignis-Variable <Name> in Task <Name> muss vom Typ BOOL sein.**

Verwenden Sie eine Variable vom Typ BOOL als Ereignis-Variable.

3554 **Taskeintrag <Name> muss ein Programm oder eine globale Funktionsblockinstanz sein.**

Sie haben im Feld [Programmaufruf] eine Funktion oder einen nicht definierten Baustein eingetragen.

3555 **Der Taskeintrag <Name> ist falsch parametriert.**

Sie haben im Feld [Programmaufruf] Parameter angegeben, die nicht der Deklaration des Bausteins entsprechen.

3556 **Tasks werden vom eingestellten Target nicht unterstützt.**

Für das aktuell eingestellte Zielsystem ist die im Projekt vorliegende Taskkonfiguration nicht möglich. Wählen Sie ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3557 **Die maximale Anzahl von Tasks (<Anzahl>) wurde überschritten.**

Das aktuell eingestellte Zielsystem erlaubt die momentan in der Taskkonfiguration vorliegende Anzahl von Tasks nicht. Wählen SIe ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3558 **Priorität der Task <Name> liegt außerhalb des gültigen Bereichs zwischen <Untergrenze> und <Obergrenze>.**

Das aktuell eingestellte Zielsystem erlaubt die momentan in der Taskkonfiguration vorgegebene Priorität der Task nicht. Wählen SIe ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3559 **Task <Name>: Intervalltask wird vom aktuellen Zielsystem nicht unterstützt.**

Das aktuell eingestellte Zielsystem erlaubt die momentan in der Taskkonfiguration definierte Intervall-Task nicht. Wählen Sie ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3560 **Task <Name>: Freilaufende Tasks werden vom aktuellen Zielsystem nicht unterstützt.**

Das aktuell eingestellte Zielsystem erlaubt die momentan in der Taskkonfiguration definierten freilaufenden Tasks nicht. Wählen Sie ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3561 **Task <Name>: Ereignistasks werden vom aktuellen Zielsystem nicht unterstützt.**

Das aktuell eingestellte Zielsystem erlaubt die momentan in der Taskkonfiguration definierten ereignisgesteuerten Tasks nicht. Wählen Sie ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3562 **Task <Name>: Extern ereignisgesteuerte Tasks werden vom aktuellen Zielsystem nicht unterstützt.**

Das aktuell eingestellte Zielsystem erlaubt die momentan in der Taskkonfiguration definierten, extern ereignisgesteuerten Tasks nicht. Wählen Sie ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3563 **Das Intervall von Task <Name> liegt außerhalb des gültigen Bereichs von <Bereichsgrenze> bis <Bereichsgrenze>.**

Korrigieren Sie die Angabe des Intervalls im Dialog [Eigenschaften] für diese Task in der Taskkonfiguration.

3564 **Das externe Ereignis <Name> der Task <Name> wird vom aktuellen Zielsystem nicht unterstützt.**

Das aktuell eingestellte Zielsystem erlaubt das momentan für die Task definierte externe Ereignis nicht. Passen Sie die Taskkonfiguration an.

3565 **Die maximale Anzahl von Ereignistasks von <Anzahl> wurde überschritten.**

Das aktuell eingestellte Zielsystem erlaubt nicht so viele Ereignistasks wie momentan definiert. Wählen Sie ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3566 **Die maximale Anzahl von Intervalltasks von <Anzahl> wurde überschritten.**

Das aktuell eingestellte Zielsystem erlaubt nicht so viele Intervalltasks wie momentan definiert. Wählen Sie ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3567 **Die maximale Anzahl von frei laufenden Tasks von <Anzahl> wurde überschritten.**

Das aktuell eingestellte Zielsystem erlaubt nicht so viele freilaufende Tasks wie momentan definiert. Wählen Sie ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3568 **Die maximale Anzahl von externen Ereignistasks von <Anzahl> wurde überschritten.**

Das aktuell eingestellte Zielsystem erlaubt nicht so viele externe Tasks wie momentan definiert. Wählen Sie ein geeignetes Zielsystem oder passen Sie die Taskkonfiguration an.

3569 **Baustein <Name> für Systemereignis <Name> nicht definiert.**

Der genannte Baustein, der vom definierten Systemereignis in der Taskkonfiguration aufgerufen wird, ist im Projekt nicht verfügbar. Passen Sie die Taskkonfiguration an bzw. stellen Sie sicher, dass der aufzurufende Baustein unter diesem Namen im Projekt vorhanden ist.

3570 **Die Tasks <Name1> und <Name2> haben die gleiche Priorität.**

Verändern Sie in der Taskkonfiguration die Prioritätsstufe der Tasks.

3571 **Die Bibliothek "SysLibCallback" ist nicht eingebunden! System-Ereignisse können nicht generiert werden.**

Für die Verwendung von ereignisgesteuerten Tasks (Event Tasks) ist die Bibliothek `SysLibCallback.lib` erforderlich. Binden Sie die Bibliothek ein oder verwenden Sie in der Taskkonfiguration keine Task vom Typ "ereignisgesteuert".

3572 **Das Watchdog-Intervall von Task <Name> liegt außerhalb des gültigen Bereichs von <Zahl> µs bis <Zahl> µs.**

In der Taskkonfiguration im Dialog [Taskeigenschaften] ist für die Watchdog-Zeit ein Mikrosekunden-Wert eingetragen, der außerhalb des in der XML-Beschreibungsdatei definierten Bereichs liegt.

3573 **Das Watchdog-Intervall von Task <Name> liegt außerhalb des gültigen Bereichs von <Zahl>% bis <Zahl>%.**

In der Taskkonfiguration im Dialog [Taskeigenschaften] ist für die Watchdog-Zeit ein Prozentwert eingetragen, der außerhalb des in der XML-Beschreibungsdatei definierten Bereichs liegt.

3574 **Die Event-Variable <Name> bzw. deren direkte Adresse darf nur einmal als Event verwendet werden.**

Ein Singleton-Event wird in der Taskkonfiguration mehrmals verwendet. Lesen Sie hierzu die Beschreibung zu den Menübefehlen [Einfügen] / [Task einfügen] oder [Einfügen] / [Task anhängen].

3600 **Implizite Variable nicht gefunden.**

Wenden Sie zunächst den Befehl [Alles übersetzen] an. Falls die Fehlermeldung erneut erscheint, wenden Sie sich bitte an Ihren Steuerungshersteller.

3601 **<Name> ist ein reservierter Variablenname.**

Sie haben im Projekt eine Variable deklariert, die bereits für den Codegenerator reserviert ist. Benennen Sie diese Variable um.

3610 **<Name> wird nicht unterstützt.**

Das angegebene Feature wird von dieser Version nicht unterstützt.

3611 **Das Übersetzungsverzeichnis <Name> ist ungültig.**

Sie haben in den Projektoptionen, Kategorie [Verzeichnisse], ein ungültiges Verzeichnis für die Übersetzungsdateien eingetragen.

3612 **Maximale Anzahl der Bausteine (<Anzahl>) überschritten! Übersetzung wird abgebrochen.**

Sie verwenden zu viele Bausteine und Datentypen im Projekt. Verändern Sie die max. Anzahl der Bausteine in den Zielsystemeinstellungen, Kategorie [Speicheraufteilung].

3613 **Übersetzung abgebrochen.**

Die Übersetzung wurde durch den Benutzer abgebrochen.

3614 **Das Projekt enthält keinen Baustein <Name> (Einsprungfunktion) oder eine Taskkonfiguration.**

Ein Projekt benötigt eine Einsprungfunktion vom Typ `Programm` (z. B. `PLC_PRG`) oder eine Taskkonfiguration.

3615 **<Name> (Einsprungfunktion) muss vom Typ Programm sein.**

Sie verwenden eine Einsprungsfunktion (z. B. `PLC_PRG`), die nicht vom Typ `Programm` ist.

3616 **Programme in externen Bibliotheken werden nicht unterstützt.**

Die zu speichernde Bibliothek enthält ein Programm, das bei der Verwendung der Bibliothek nicht zur Verfügung steht.

3617 **Zu wenig Speicher.**

Erhöhen Sie die virtuelle Speicherkapazität Ihres Rechners.

3618 **Bit-Zugriffe werden vom aktuellen Codegenerator nicht unterstützt.**

Der Codegenerator für das aktuell eingestellte Zielsystem unterstützt keine Bitzugriffe auf Variablen.

3619 **Objekt-Datei <Name> und Bibliothek <Name> haben unterschiedliche Version!**

Stellen Sie sicher, dass für die Bibliothek zueinander passende Versionen der Dateien `*.lib` und `*.obj` bzw. `*.hex` vorliegen. Die Dateien müssen den exakt gleichen Zeitstempel tragen.

3620 **Der Baustein PLC_PRG darf in einer Bibliothek nicht vorhanden sein.**

Sie wollen das Projekt als Bibliothek der Version 2.1. speichern. In dieser Version darf eine Bibliothek keinen Baustein `PLC_PRG` enthalten. Verwenden Sie einen anderen Bausteinnamen.

3621 **Die Übersetzungsdatei <Name> kann nicht beschrieben werden.**

Im Pfad, der für die genannte Übersetzungsdatei vorgesehen ist (→ Kap. 4.2.7, [Verzeichnisse] ([Directories])), liegt vermutlich bereits eine gleichnamige Datei vor, die schreibgeschützt ist. Entfernen Sie die Datei bzw. ändern Sie die Zugriffsrechte.

3622 **Die Symboldatei <Name> konnte nicht erzeugt werden.**

Im Pfad, der für die Symboldatei vorgesehenen ist (üblicherweise Projektverzeichnis), liegt vermutlich bereits eine gleichnamige Datei vor, die schreibgeschützt ist. Entfernen Sie die Datei bzw. ändern Sie die Zugriffsrechte.

3623 **Die Bootprojektdatei <Name> kann nicht beschrieben werden.**

In dem vom Zielsystem abhängigen Pfad, der für die Symboldatei vorgesehenen ist, liegt vermutlich bereits eine gleichnamige Datei vor, die schreibgeschützt ist. Entfernen Sie die Datei bzw. ändern Sie die Zugriffsrechte.

3700 **Ein Baustein mit Namen <Name> ist bereits in Bibliothek <Name>.**

Sie verwenden einen Bausteinnamen, der bereits für einen Bibliotheksbaustein vergeben ist. Benennen Sie den Baustein um.

3701 **Der Bausteinname im Deklarationsnamen stimmt nicht mit dem in der Objektliste überein.**

Benennen Sie Ihren Baustein über den Menübefehl [Projekt] / [Objekt umbenennen] ([Project] / [Objekt umbenennen]) neu, oder ändern Sie den Bausteinnamen in dessen Deklarationsteil. Der Name muss direkt nach den Schlüsselwörtern `PROGRAM`, `FUNCTION` oder `FUNCTIONBLOCK` stehen.

3702 **Zu viele Bezeichner.**

Pro Variablendeklaration können maximal 100 Bezeichner angegeben werden.

3703 **Mehrere Deklarationen mit dem gleichen Bezeichner <Name>.**

Im Deklarationsteil des Objekts existieren mehrere Bezeichner mit dem gleichen Namen.

3704 **Datenrekursion: <Baustein 0> -> <Baustein 1> -> .. -> <Baustein 0>.**

Eine FB-Instanz wurde verwendet, die sich selbst wieder benötigt.

3705 **<Name>: VAR_IN_OUT in Top-Level-Baustein nicht erlaubt, wenn keine Taskkonfiguration vorhanden ist.**

Setzen Sie eine Taskkonfiguration auf oder stellen Sie sicher, dass im Baustein `PLC_PRG` keine Variablen vom Typ `VAR_IN_OUT` verwendet werden.

3706 **Modifizierer CONSTANT ist nur für VAR, VAR_INPUT, VAR_EXTERNAL und VAR_GLOBAL zulässig.**

Für Variablen vom Typ `VAR`, `VAR_INPUT` können keine Konstanten deklariert werden.

3720 **Nach AT muss eine Adresse stehen.**

Fügen Sie eine gültige Adresse nach dem Schlüsselwort `AT` ein, oder ändern Sie das Schlüsselwort `AT`.

3721 **Nur VAR und VAR_GLOBAL dürfen auf Adressen gelegt werden.**

Kopieren Sie die Deklaration in einen Bereich vom Typ `VAR` oder `VAR_GLOBAL`.

3722 **Auf der angegebenen Adresse dürfen nur einfache boolesche Variablen stehen.**

Ändern Sie die Adresse oder den in der Deklaration angegebenen Typ der Variablen.

3726 **Konstante kann nicht auf direkte Adresse gelegt werden.**

Verändern Sie die Adresszuweisung.

3727 **Auf diese Adresse dürfen keine Arrays gelegt werden.**

Verändern Sie die Adresszuweisung entsprechend.

3728 **Unzulässige Adresse: <Adresse>.**

Diese Adresse wird von der Steuerungskonfiguration nicht unterstützt. Prüfen Sie die Steuerungskonfiguration bzw. korrigieren Sie die Adresse.

3729 **Unzulässiger Typ <Name> auf Adresse: <Name>.**

Der Typ dieser Variable kann auf der angegebenen Adresse nicht platziert werden.

Beispiel

Für ein Zielsystem, das mit Alignment 2 arbeitet, ist die Deklaration
`var1 AT %IB1:WORD;` ungültig.

3740 **Unbekannter Typ: <Name>.**

Sie verwenden in der Variablendeklaration einen ungültigen Typen.

3741 **Typbezeichner erwartet.**

Sie verwenden ein Schlüsselwort oder einen Operator anstelle eines gültigen Typbezeichners.

3742 **Aufzählungswert erwartet.**

In der Definition des Aufzählungstyps fehlt ein Bezeichner nach der öffnenden Klammer oder nach einem Komma innerhalb des Klammerbereichs.

3743 **Ganze Zahl erwartet.**

Aufzählungswerte können nur mit ganzen Zahlen vom Typ `INT` initialisiert werden.

3744 **Enum-Konstante <Name> bereits definiert.**

Prüfen Sie, ob Sie folgende Regeln bei der Vergabe von Enumerationswerten beachtet haben:

- Innerhalb einer `ENUM`-Definition müssen alle Werte eindeutig sein.
- Innerhalb aller globalen `ENUM`-Definitionen müssen alle Werte eindeutig sein.
- Innerhalb aller lokalen `ENUM`-Definitionen eines Bausteins müssen alle Werte eindeutig sein.

3745 **Bereichsgrenzen sind nur für Integer-Datentypen erlaubt!**

Unterbereichstypen können nur auf Basis von Integer-Datentypen definiert werden.

3746 **Bereichsgrenze <Name> nicht kompatibel zu Datentyp <Name>.**

Eine Grenze des für den Unterbereichstypen angegebenen Bereichs liegt außerhalb der für den Basistypen erlaubten Grenzen.

3747 **Unbekannte Stringlänge: <Name>.**

Sie verwenden eine unbekannte Konstante zur Definition der Stringlänge.

3748 **Mehr als 3 Dimensionen sind für ein Array unzulässig.**

Sie verwenden mehr als die zulässigen drei Dimensionen für ein Array. Verwenden Sie gegebenenfalls ein `ARRAY OF ARRAY`.

3749 **Untergrenze <Name> unbekannt.**

Sie verwenden eine nicht definierte Konstante als Untergrenze eines Unterbereichs- oder Array-Typen.

3750 **Obergrenze <Name> unbekannt.**

Sie verwenden eine nicht definierte Konstante als Obergrenze eines Unterbereichs- oder Array-Typen.

3751 **Ungültige Stringlänge <Anzahl Zeichen>.**

Die im eingestellten Zielsystem erlaubte maximale Stringlänge (Anzahl Zeichen) wird in der vorliegenden Deklaration überschritten.

3752 **Mehr als 9 Dimensionen sind für geschachtelte Arrays unzulässig.**

Ein Array kann max. dreidimensional sein. Die durch Schachtelung von Arrays (z. B. `arr: ARRAY [0..2,0..2,0..2] OF ARRAY [0..2,0..2,0..2] OF ARRAY [0..2,0..2,0..2, 0..2] OF DINT`) erreichte Dimension darf maximal "9" sein und wird im vorliegenden Fall überschritten. Reduzieren Sie entsprechend auf maximal neun Dimensionen.

3760 **Fehlerhafter Initialwert.**

Verwenden Sie einen Initialwert, welcher der Typdefinition entspricht. Verwenden Sie dazu den Variablendeklarationsdialog (Tastaturbefehl <Umschalt>+<F2> oder den Menübefehl [Bearbeiten] / [Variablendeklaration]).

3761 **VAR_IN_OUT Variablen dürfen keinen Initialwert haben.**

Entfernen Sie die Initialisierung bei der Deklaration der Variablen.

3780 **VAR, VAR_INPUT, VAR_OUTPUT oder VAR_IN_OUT erwartet.**

Die erste Zeile nach dem Namen eines Bausteins muss eines dieser Schlüsselwörter beinhalten.

3781 **END_VAR oder Bezeichner erwartet.**

Schreiben Sie einen gültigen Bezeichner oder `END_VAR` an den Anfang der Deklarationszeile.

3782 **Unerwartetes Ende.**

Im Deklarationsteil: Fügen Sie gegebenenfalls am Ende des Deklarationsteils das Schlüsselwort END_VAR ein. Die Fehlermeldung wird auch im Zusammenhang mit Fehler 3703 ausgegeben, wenn zwei gleiche Deklarationen am Ende des Deklarationsteils stehen.

Im Texteditor: Fügen Sie Anweisungen ein, die die letzte Anweisungssequenz beenden (z. B. END_IF).

3783 **END_STRUCT oder Bezeichner erwartet.**

Stellen Sie sicher, dass die Typdeklaration richtig abgeschlossen ist.

3784 **Das Attribut <Name> wird vom Zielsystem nicht unterstützt.**

Das Zielsystem unterstützt diesen Variablentyp nicht (z. B. RETAIN, PERSISTENT).

3800 **Die globalen Variablen brauchen zu viel Speicher. Erhöhen Sie den verfügbaren Speicher in den Projektoptionen.**

Vergrößern Sie in den Projektoptionen, Kategorie [Übersetzungsoptionen] ([Build]) die eingestellte Anzahl der Segmente.

3801 **Die Variable <Name> ist zu groß. (<Größe> Byte)**

Die Variable verwendet einen Typen, der größer als ein Datensegment ist.

Die Segmentgröße ist vom Zielsystem abhängig und lässt sich in den Zielsystemeinstellungen in der Kategorie [Speicheraufteilung] einstellen. Sollten Sie dort keine Eingabemöglichkeit finden, wenden Sie sich bitte an Ihren Steuerungshersteller.

3802 **Speicher für Retainvariable aufgebraucht. Variable <Name>, <Anzahl> Byte.**

Der verfügbare Speicherplatz für die Variable <Name> vom Typ RETAIN ist aufgebraucht.

Der Speicherplatz ist vom Zielsystem abhängig und lässt sich in den Zielsystemeinstellungen in der Kategorie [Speicheraufteilung] einstellen. Sollten Sie dort keine Eingabemöglichkeit finden, wenden Sie sich bitte an Ihren Steuerungshersteller.

Beachten Sie auch, dass bei Instanzen von Funktionsblöcken, in denen eine Variable vom Typ RETAIN verwendet wird, die gesamte Instanz im RETAIN-Speicher verwaltet wird!

3803 **Speicher für globale Variablen aufgebraucht. Variable <Name>, <Anzahl> Byte.**

Der verfügbare Speicherplatz für globale Variablen ist erschöpft.

Der Speicherplatz ist vom Zielsystem abhängig und lässt sich in den Zielsystemeinstellungen in der Kategorie [Speicheraufteilung] einstellen. Sollten Sie dort keine Eingabemöglichkeit finden, wenden Sie sich bitte an Ihren Steuerungshersteller.

3820 **VAR_OUTPUT und VAR_IN_OUT ist in Funktionen nicht erlaubt.**

Sie dürfen in einer Funktion keine Ausgangs- bzw. Referenzparameter definieren.

3821 **Zumindest ein Input bei einer Funktion erforderlich.**

Fügen Sie mindestens einen Eingabeparameter für die Funktion ein.

3840 **Unbekannte globale Variable <Name>!**

Sie verwenden im Baustein eine Variable vom Typ VAR_EXTERNAL, für die keine entsprechende globale Variable deklariert ist.

3841 **Deklaration von <Name> stimmt nicht mit globaler Deklaration überein!**

Die Typangabe in der Deklaration der Variable vom Typ VAR_EXTERNAL stimmt nicht mit der in der globalen Deklaration überein.

3900 **Mehrfache Unterstriche im Bezeichner.**

Entfernen Sie mehrfache Unterstriche im Bezeichner.

3901 **Es sind maximal 4 Adressfelder zulässig.**

Sie verwenden eine direkte Adresszuweisung auf eine Adresse, die mehr als vier Stufen enthält (z. B. `%QB0.1.1.0.1`).

3902 **Schlüsselwörter müssen groß geschrieben werden.**

Schreiben Sie das Schlüsselwort in Großbuchstaben bzw. aktivieren Sie die Option [Automatisch formatieren].

3903 **Ungültige Zeitkonstante.**

Das Format der Konstante entspricht nicht IEC 61131-3.

3904 **Überlauf in Zeitkonstante.**

Sie verwenden einen Wert für die Zeitkonstante, der im internen Format nicht mehr darstellbar ist. Der maximal darstellbare Wert ist `t#49d17h2m47s295ms`.

3905 **Ungültige Datumskonstante.**

Das Format der Konstante entspricht nicht IEC 61131-3.

3906 **Ungültige Tageszeitkonstante.**

Das Format der Konstante entspricht nicht IEC 61131-3.

3907 **Ungültige Datum/Zeit-Konstante.**

Das Format der Konstante entspricht nicht IEC 61131-3.

3908 **Ungültige Stringkonstante.**

Die Stringkonstante enthält ein ungültiges Zeichen.

4000 **Bezeichner erwartet.**

Geben Sie an dieser Stelle einen gültigen Bezeichner an.

4001 **Variable <Name> nicht deklariert.**

Deklarieren Sie die Variable lokal oder global.

4010 **Unverträgliche Typen: Kann <Name> nicht in <Name> konvertieren.**

Prüfen Sie die erforderlichen Typen des Operators und ändern Sie den Typ der Variablen, die den Fehler produziert hat, oder wählen Sie eine andere Variable.

4011 **Unzulässiger Typ in Parameter <Parameter> von <Name>: Kann <Name> nicht in <Name> konvertieren.**

Der Typ des Aktual-Parameters kann nicht in den des Formal-Parameters überführt werden. Verwenden Sie eine Typkonvertierung oder verwenden Sie einen entsprechenden Variablentypen.

4012 **Unzulässiger Typ für Eingang <Name> von <Name>: Kann <Name1> nicht in <Name2> konvertieren.**

Der Variablen `<Name>` wird ein Wert mit dem unzulässigen Typ `<Name2>` zugewiesen. Ändern Sie die Variable oder die Konstante zu einer Variablen oder zu einer Konstanten mit dem Typ `<Name1>` oder verwenden Sie eine Typkonvertierung bzw. eine Konstante mit Typ-Präfix.

4013 **Unzulässiger Typ für Ausgang <Name> von <Name>: Kann <Name1> nicht in <Name2> konvertieren.**

Der Variablen `<Name>` wird ein Wert mit dem unzulässigen Typ `<Name2>` zugewiesen. Ändern Sie die Variable oder die Konstante zu einer Variablen oder Konstanten mit dem Typ `<Name1>` oder verwenden Sie eine Typkonvertierung bzw. eine Konstante mit Typ-Präfix.

4014 **Konstante mit Typ-Präfix: <Name> kann nicht nach <Name> konvertiert werden.**

Der Typ der Konstanten ist nicht kompatibel mit dem Typen des Präfix.

Beispiel: `SINT#255`

4015 **Unzulässiger Datentyp <Name> für direkten Bitzugriff**

Direkte Bitadressierung ist nur für Integer- und Bitstring-Datentypen zulässig. Sie verwenden im Bit-Zugriff `<var1>.<bit>` eine Variable `var1` vom Typ `REAL`/`LREAL` oder eine Konstante.

4016 **Bit-Index <Zahl> außerhalb des gültigen Bereichs für Variable des Typs <Name>.**

Sie versuchen, auf ein Bit zuzugreifen, das für den Datentyp der Variable nicht definiert ist.

4017 **MOD ist für REAL nicht definiert.**

Der Operator `MOD` kann nur für Integer- und Bitstring-Datentypen verwendet werden.

4020 **Operanden von ST, STN, S, R müssen Variable mit Schreibzugriff sein.**

Ersetzen Sie den ersten Operanden durch eine Variable, auf die geschrieben werden kann.

4021 **Kein Schreibzugriff auf <Name>.**

Ersetzen Sie die Variable durch eine mit Schreibzugriff.

4022 **Operand erwartet.**

Ergänzen Sie einen Operanden hinter dem bestehenden Befehl.

4023 **Nach "+" bzw. "-" wird eine Zahl erwartet.**

Geben Sie eine Zahl ein.

4024 **Erwarte <Operator 0> oder <Operator 1> oder ... vor <Name>.**

Geben Sie an der genannten Stelle einen gültigen Operator ein.

4025 **Erwarte ":=" oder "=>" vor <Name>.**

Geben Sie an der genannten Stelle einen der beiden Operatoren ein.

4026 **BITADR erwartet eine Bitadresse oder eine Variable auf einer Bitadresse.**

Verwenden Sie eine gültige Bitadresse (z. B. `%IX0.1`).

4027 **Ganze Zahl oder symbolische Konstante erwartet.**

Fügen Sie eine ganze Zahl oder den Bezeichner einer gültigen Konstante ein.

4028 **INI Operator benötigt eine Funktionsblockinstanz oder eine Strukturvariable.**

Prüfen Sie den Typen der Variablen, auf den Sie den INI-Operator anwenden.

4029 **Ineinander verschachtelte Aufrufe derselben Funktion sind nicht möglich.**

Bei nicht reentranten Zielsystemen und im Simulationsmodus darf ein Funktionsaufruf keinen Aufruf auf sich selbst als Parameter enthalten.

Beispiel: `fun1(a,fun1(b,c,d),e);`

Verwenden Sie eine Zwischenvariable.

4030 **Als Operanden zu ADR sind keine Konstanten und Ausdrücke erlaubt.**

Ersetzen Sie die Konstante oder den Ausdruck durch eine Variable oder eine direkte Adresse.

4031 **Der Adressoperator ist auf Bits nicht erlaubt! Verwenden Sie stattdessen BITADR.**

Verwenden Sie `BITADR`. Beachten Sie: Der `BITADR` liefert keine physikalische Speicheradresse.

4032 **<Anzahl> Operanden sind zu wenige für <Name>. Es werden mindestens <Anzahl> benötigt.**

Überprüfen Sie, wie viele Operanden der Operator `<Name>` benötigt, und fügen Sie die fehlenden ein.

4033 **<Anzahl> Operanden sind zu viele für <Name>. Es werden genau <Anzahl> benötigt.**

Überprüfen Sie, wie viele Operanden der Operator `<Name>` benötigt, und entfernen Sie die überzähligen.

4034 **Division durch 0.**

Sie verwenden eine Division durch 0 in einem konstanten Ausdruck. Verwenden Sie gegebenenfalls eine Variable mit dem Wert 0 um einen Laufzeitfehler zu erzwingen.

4035 **ADR darf nicht auf VAR CONSTANT angewendet werden, wenn "Konstanten ersetzen" aktiviert ist.**

Ein Adresszugriff auf Konstanten, für die die direkten Werte verwendet werden, ist nicht möglich. Deaktivieren Sie gegebenenfalls die Option [Konstanten ersetzen] in den Projektoptionen, Kategorie [Übersetzungsoptionen].

4040 **Sprungmarke <LabelName> ist nicht definiert.**

Definieren Sie eine Marke mit dem Namen <LabelName> oder ändern Sie <LabelName> in eine definierte Marke.

4041 **Mehrfache Definition der Sprungmarke <Name>.**

Die Sprungmarke <Name> ist im Baustein mehrfach definiert. Benennen Sie die Sprungmarke entsprechend um oder entfernen Sie die Definition.

4042 **Es dürfen höchstens <Anzahl> Sprungmarken in Folge stehen.**

Die Anzahl der Sprungmarken pro Anweisung ist auf <Anzahl> begrenzt. Fügen Sie eine Dummy-Anweisung ein.

4043 **Labelformat ungültig. Ein Label muss ein Bezeichner sein, dem ein Doppelpunkt folgen kann.**

Der für das Label verwendete Name ist entweder kein gültiger Bezeichner oder es fehlt der Doppelpunkt bei der Definition.

4050 **Baustein <Name> existiert nicht im Projekt.**

Definieren Sie einen Baustein mit dem Namen <Name> durch die Menübefehle [Projekt] / [Objekt anfügen] oder ändern Sie <Name> in den Namen eines definierten Bausteins.

4051 **<Name> ist keine Funktion.**

Verwenden Sie für <Name> einen der im Projekt oder den Bibliotheken definierten Funktionsnamen.

4052 **<Instanzname> muss eine deklarierte Instanz des Funktionsblocks <Name> sein.**

Verwenden Sie für <Instanzname> eine im Projekt definierte Instanz des Typs <Name> oder ändern Sie den Typen von <Instanzname> in <Name>.

4053 **<Name> ist kein gültiger Baustein oder Operator.**

Ersetzen Sie <Name> durch den Namen eines im Projekt definierten Bausteins oder eines Operators.

4054 **Bausteinname als Parameter von INDEXOF erwartet.**

Der angegebene Parameter ist kein gültiger Bausteinname.

4060 **VAR_IN_OUT Parameter <Name> von <Name> benötigt Variable mit Schreibzugriff als Eingabe.**

An VAR_IN_OUT-Parameter müssen Variablen mit Schreibzugriff übergeben werden, damit diese innerhalb des Bausteins modifiziert werden können.

4061 **VAR_IN_OUT Parameter <Name> von <Name> muss belegt werden.**

An VAR_IN_OUT-Parameter müssen Variablen mit Schreibzugriff übergeben werden, damit diese innerhalb des Bausteins modifiziert werden können.

4062 **Kein Zugriff auf VAR_IN_OUT Parameter <Name> von <Name> von außen.**

VAR_IN_OUT-Parameter dürfen nur innerhalb des Bausteins beschrieben oder gelesen werden, da es sich um eine Übergabe durch Referenz handelt.

4063 **VAR_IN_OUT Parameter <Name> von <Name> kann nicht mit Bitadressen belegt werden.**

Eine Bitadresse ist keine gültige physikalische Adresse. Übergeben Sie eine Variable oder eine direkte Nicht-Bitadresse.

4064 **VAR_IN_OUT darf in lokalem Aktionsaufruf nicht überschrieben werden!**

Löschen Sie die Belegung der VAR_IN_OUT-Variablen für den lokalen Aktionsaufruf.

4070 **Ein Baustein enthält einen zu tief geschachtelten Ausdruck.**

Verkleinern Sie die Schachtelungstiefe, indem Sie mit Hilfe von Zuweisungen auf Zwischenvariablen den Ausdruck auf mehrere Ausdrücke umverteilen.

4071 **Netzwerk ist zu groß.**

Teilen Sie das Netzwerk in mehrere Netzwerke auf.

4072 **Inkonsistente Benutzung des Aktionsbezeichners in FB-Typ <Name> und Instanz <Name>.**

Sie haben zwei Aktionen eines Funktionsblocks FB definiert, z. B. a1 und a2, verwenden beim Aufruf einer der Aktionen im FUP jedoch bei der Typangabe (Bezeichnung innerhalb der Box) einen anderen Aktionsbezeichner (z. B. fb.a1) als in der Instanz-Bezeichnung (z. B. inst.a2, oberhalb der Box). Korrigieren Sie entsprechend den Bezeichner der gewünschten Aktion.

4100 **"^" benötigt einen Pointertyp.**

Sie versuchen eine Variable zu dereferenzieren, die nicht als POINTER TO deklariert ist.

4110 **[<index>] ist nur für Arrayvariablen zulässig.**

Sie verwenden [<index>] für eine Variable, die nicht als ARRAY OF deklariert ist.

4111 **Der Ausdruck im Index eines Arrays muss ein Ergebnis vom Typ INT haben.**

Verwenden Sie einen Ausdruck des entsprechenden Typs oder eine Typkonvertierung.

4112 **Zu viele Array-Indizes.**

Überprüfen Sie die Zahl der Indizes (1, 2 oder 3), für die das Array deklariert ist, und entfernen Sie die überzähligen.

4113 **Zu wenig Array-Indizes.**

Überprüfen Sie die Zahl der Indizes (1, 2 oder 3), für die das Array deklariert ist, und ergänzen Sie die fehlenden.

4114 **Ein konstanter Index liegt nicht im Array-Bereich.**

Stellen Sie sicher, dass die verwendeten Indizes innerhalb der Grenzen des Arrays liegen.

4120 **Vor dem "." muss eine Strukturvariable stehen.**

Der Bezeichner links vom Punkt muss entweder eine Variable vom Typ `STRUCT` bzw. `FUNCTION_BLOCK` oder der Name einer Funktion (`FUNCTION`) bzw. eines Programms (`PROGRAM`) sein.

4121 **<Name> ist keine Komponente von <Objektname>.**

Die Komponente <`Name`> ist in der Definition des Objekts <`Objektname`> nicht enthalten.

4122 **<Name> ist kein Eingabeparameter des aufgerufenen Funktionsblocks.**

Überprüfen Sie die Eingabevariablen des aufgerufenen Funktionsblocks und ändern Sie <`Name`> in eine dieser Variablen.

4200 **LD erwartet.**

Fügen Sie im Editorfenster des AWL-Bausteins bzw. nach der Sprungmarke zumindest eine `LD`-Anweisung ein.

4201 **AWL-Operator erwartet.**

Jede AWL-Anweisung muss mit einem Operator oder einer Sprungmarke beginnen.

4202 **Unerwartetes Ende des Klammerausdrucks.**

Fügen Sie die schließende Klammer ein.

4203 **<Name> in Klammern ist nicht zulässig.**

Der angegebene Operator ist innerhalb eines AWL-Klammerausdrucks nicht zulässig.

Nicht zulässig sind: `JMP`, `RET`, `CAL`, `LDN`, `LD`, `TIME`

4204 **Schließende Klammer ohne zugehörige öffnende Klammer.**

Fügen Sie die öffnende Klammer ein oder löschen Sie die schließende.

4205 **Nach ")" ist kein Komma zulässig.**

Entfernen Sie das Komma nach der schließenden Klammer.

4206 **Keine Sprungmarken innerhalb von Klammerausdrücken.**

Verschieben Sie die Sprungmarke, sodass sie außerhalb des Klammerausdrucks liegt.

4207 **N-Modifikator verlangt einen Operanden vom Typ BOOL, BYTE, WORD or DWORD.**

Der N-Modifikator benötigt einen Datentypen, für den eine boolesche Negation ausgeführt werden kann.

4208 **Der Ausdruck vor einem bedingten Befehl muss ein Ergebnis vom Typ BOOL liefern.**

Stellen Sie sicher, dass der Ausdruck ein boolesches Ergebnis liefert oder verwenden Sie eine Typkonvertierung.

4209 **An dieser Stelle ist kein Funktionsname zulässig.**

Tauschen Sie den Funktionsaufruf gegen eine Variable oder eine Konstante aus.

4210 **CAL, CALC und CALN benötigen eine Funktionsblockinstanz als Operanden.**

Deklarieren Sie eine Instanz des Funktionsblocks, den Sie aufrufen möchten.

4211 **Kommentar ist in AWL nur am Zeilenende zulässig.**

Verschieben Sie den Kommentar an das Zeilenende oder in eine eigene Zeile.

4212 **Akkumulator ist ungültig vor bedingter Anweisung.**

Der Inhalt des Akkumulators ist nicht definiert. Dies ist der Fall nach Anweisungen, die kein Ergebnis liefern (z. B. CAL).

4213 **"S" und "R" verlangen einen Operanden vom Typ BOOL**

Verwenden Sie an dieser Stelle eine boolesche Variable.

4250 **Kein korrekter Anfang für eine ST Anweisung.**

Die Zeile beginnt nicht mit einer gültigen ST-Anweisung.

4251 **Funktion <Name> hat zu viele Parameter.**

Es wurden mehr Parameter angegeben, als in der Funktionsdefinition deklariert sind.

4252 **Funktion <Name> hat zu wenige Parameter.**

Es wurden weniger Parameter angegeben, als in der Funktionsdefinition deklariert sind.

4253 **IF und ELSIF benötigen als Bedingung einen booleschen Ausdruck.**

Stellen Sie sicher, dass die Bedingung ein boolescher Ausdruck ist.

4254 **WHILE benötigt als Bedingung einen booleschen Ausdruck.**

Stellen Sie sicher, dass die Bedingung ein boolescher Ausdruck ist.

4255 **UNTIL benötigt als Bedingung einen booleschen Ausdruck.**

Stellen Sie sicher, dass die Bedingung ein boolescher Ausdruck ist.

4256 **NOT verlangt einen booleschen Operanden.**

Stellen Sie sicher, dass nach `NOT` ein boolescher Ausdruck steht.

4257 **Der Zähler der FOR Anweisung muss vom Typ INT sein.**

Stellen Sie sicher, dass die Zählvariable ein Integer- oder Bitstring-Datentyp ist, z. B. `DINT`, `DWORD`.

4258 **Der Zähler in der FOR-Anweisung ist keine Variable mit Schreibzugriff.**

Ersetzen Sie die Zählvariable durch eine Variable mit Schreibzugriff.

4259 **Der Startwert der FOR-Anweisung muss vom Typ INT sein.**

Der Startwert der `FOR`-Anweisung muss kompatibel mit dem Typen der Zählvariable sein.

4260 **Der Endwert der FOR-Anweisung muss vom Typ INT sein.**

Der Endwert der `FOR`-Anweisung muss kompatibel mit dem Typen der Zählvariable sein.

4261 **Der Inkrementationswert der FOR-Anweisung muss vom Typ INT sein.**

Der Inkrementationswert der `FOR`-Anweisung muss kompatibel mit dem Typen der Zählvariable sein.

4262 **EXIT ist nur innerhalb einer Schleife erlaubt.**

Verwenden Sie `EXIT` nur innerhalb von `FOR`-, `WHILE`- oder `UNTIL`-Anweisungen.

4263 **Zahl, ELSE oder END_CASE erwartet.**

Innerhalb einer `CASE`-Anweisung können nur eine Zahl, eine `ELSE`-Anweisung oder die Endanweisung `END_CASE` angegeben werden.

4264 **Der Selector der CASE-Anweisung muss vom Typ INT sein.**

Stellen Sie sicher, dass der Selektor ein Integer- oder Bitstring-Datentyp ist (z. B. `DINT`, `DWORD`).

4265 **Nach "," wird eine Zahl erwartet.**

In der Aufzählung der `CASE`-Selektoren muss nach einem Komma ein weiterer Selektor angegeben werden.

4266 **Mindestens eine Anweisung ist erforderlich.**

Geben Sie eine Anweisung ein, mindestens einen Strichpunkt.

4267 **Ein Funktionsbausteinaufruf muss mit dem Namen einer Instanz beginnen.**

Der Bezeichner im Funktionsbausteinaufruf ist keine Instanz. Deklarieren Sie eine Instanz des gewünschten Funktionsbausteins bzw. verwenden Sie den Namen einer bereits deklarierten Instanz.

4268 **Es wird ein Ausdruck erwartet.**

An dieser Stelle muss ein Ausdruck eingegeben werden.

4269 **Nach ELSE-Zweig wird END_CASE erwartet.**

Schließen Sie die CASE-Anweisung nach dem ELSE-Zweig mit einem END_CASE ab.

4270 **CASE-Konstante "%ld" wird bereits verwendet.**

Ein CASE-Selektor darf innerhalb einer CASE-Anweisung nur einmal verwendet werden.

4271 **Die Untergrenze des angegebenen Bereichs ist größer als die Obergrenze.**

Korrigieren Sie die Selektoren-Bereichsgrenzen, sodass die Untergrenze nicht größer als die Obergrenze ist.

4272 **Erwarte Parameter <Name> an Stelle <Position> im Aufruf von <Name>!**

Wenn Sie die Funktionsparameter im Funktionsaufruf mit Angabe der Parameternamen vornehmen, muss die Reihenfolge der Parameter mit der in der Funktionsdefinition übereinstimmen.

4273 **CASE-Bereich <Bereichsgrenzen> überschneidet sich mit bereits verwendetem Bereich <Bereichsgrenzen>.**

Stellen Sie sicher, dass sich die Selektorenbereiche der CASE-Anweisung nicht überschneiden.

4274 **Mehrfacher ELSE-Zweig in CASE-Anweisung.**

Eine CASE-Anweisung darf nicht mehr als einen ELSE-Zweig enthalten.

4300 **Sprung bzw. Return benötigen eine boolesche Eingabe.**

Stellen Sie sicher, dass der Eingang für den Sprung bzw. die Return-Anweisung ein boolescher Ausdruck ist.

4301 **Baustein <Name> verlangt genau <Anzahl> Eingänge.**

Die Anzahl der Eingänge entspricht nicht der Anzahl der in der Bausteindefinition angegebenen VAR_INPUT- und VAR_IN_OUT-Variablen.

4302 **Baustein <Name> verlangt genau <Anzahl> Ausgänge.**

Die Anzahl der Eingänge entspricht nicht der Anzahl der in der Bausteindefinition angegebenen VAR_OUTPUT-Variablen.

4303 **<Name> ist kein Operator.**

Ersetzen Sie <Name> durch einen gültigen Operator.

4320 **Nicht boolescher Ausdruck <Name> bei Kontakt benutzt.**

Das Schaltsignal für einen Kontakt muss ein boolescher Ausdruck sein.

4321 **Nicht boolescher Ausdruck <Name> bei Spule benutzt.**

Die Ausgangsvariable einer Spule muss vom Typ `BOOL` sein.

4330 **Es wird ein Ausdruck erwartet bei Eingang EN des Bausteins <Name>.**

Beschalten Sie den Eingang `EN` des Bausteins `<Name>` mit einem Eingang oder einem Ausdruck.

4331 **Es wird ein Ausdruck erwartet bei Eingang <Anzahl> des Bausteins <Name>.**

Der Eingang des Operatorbausteins ist nicht beschaltet.

4332 **Es wird ein Ausdruck erwartet bei Eingang <Name> des Bausteins <Name>.**

Der Eingang des Bausteins ist vom Typ `VAR_IN_OUT` und ist nicht beschaltet.

4333 **Bezeichner in Sprung erwartet.**

Das angegebene Sprungziel ist kein gültiger Bezeichner.

4334 **Es wird ein Ausdruck erwartet beim Eingang des Sprungs.**

Beschalten Sie den Eingang des Sprungs mit einem booleschen Ausdruck. Wenn dieser TRUE ist, wird der Sprung ausgeführt.

4335 **Es wird ein Ausdruck erwartet beim Eingang von Return.**

Beschalten Sie den Eingang der `Return`-Anweisung mit einem booleschen Ausdruck. Wenn dieser TRUE ist, wird der Sprung ausgeführt.

4336 **Es wird ein Ausdruck erwartet beim Eingang des Ausgangs.**

Verknüpfen Sie den Ausgang mit einem Ausdruck, der diesem Ausgang zugewiesen werden kann.

4337 **Bezeichner für Eingang erwartet.**

Fügen Sie in der Eingangsbox einen gültigen Ausdruck oder Bezeichner ein.

4338 **Baustein <Name> hat keine echten Eingänge.**

Keiner der Eingänge des Operatorbausteins `<Name>` ist mit einem gültigen Ausdruck beschaltet.

4339 **Unverträgliche Typen bei Ausgang: Kann <Name> nicht in <Name> konvertieren.**

Der Ausdruck in der Ausgangsbox ist nicht typkomaptibel mit dem Ausdruck, der ihm zugewiesen werden soll.

4340 **Sprung benötigt eine boolesche Eingabe.**

Stellen Sie sicher, dass der Eingang für den Sprung ein boolescher Ausdruck ist.

4341 **Return benötigt eine boolesche Eingabe.**

Stellen Sie sicher, dass der Eingang für die `Return`-Anweisung ein boolescher Ausdruck ist.

4342 **Eingang EN der Box benötigt eine boolesche Eingabe.**

Verknüpfen Sie den EN-Eingang des Bausteins mit einem gültigen booleschen Ausdruck.

4343 **Konstantenbelegung: Unzulässiger Typ für Parameter <Name> von <Name>: Kann <Typ> nicht in <Typ> konvertieren.**

Sie haben Eingang `<Name>` von Baustein `<Name>` als `VAR_INPUT CONSTANT` deklariert. Sie haben diesem im Dialog [Parameter bearbeiten] jedoch einen Ausdruck zugewiesen, der nicht typkompatibel ist.

4344 **S und R benötigen boolesche Operanden.**

Setzen Sie hinter der Set- bzw. Reset-Anweisung einen gültigen booleschen Ausdruck ein.

4345 **Unzulässiger Typ für Parameter <Name> von <Name>: Kann <Typ> nicht in <Typ> konvertieren.**

Sie haben Eingang `<Name>` von Baustein `<Name>` einen Ausdruck zugewiesen, der nicht typkompatibel ist.

4346 **Ein Ausgang darf keine Konstante sein.**

Das Ziel einer Zuweisung muss eine Variable oder direkte Adresse mit Schreibzugriff sein.

4347 **VAR_IN_OUT-Parameter benötigt Variable mit Schreibzugriff.**

An `VAR_IN_OUT`-Parameter müssen Variablen mit Schreibzugriff übergeben werden, damit diese innerhalb des Bausteins modifiziert werden können.

4348 **Unzulässiger Programmname <Name>. Es existiert bereits eine Variable mit demselben Namen.**

Sie haben im CFC-Editor eine Programmbox eingefügt, die denselben Namen hat wie eine bereits bestehende (globale) Variable. Benennen Sie entsprechend um.

4350 **Eine AS-Aktion kann nicht von außerhalb aufgerufen werden.**

AS-Aktionen können nur innerhalb des AS-Bausteins aufgerufen werden, in dem sie definiert sind.

Die Fehlermeldung wird allerdings auch ausgegeben, wenn Sie eine Aktion zwar zulässig von innerhalb des AS-Bausteins aufrufen, aber keine IEC-Schritte verwenden und dennoch die Bibliothek `iecsfc.lib` noch im Projekt eingebunden ist. In diesem Fall entfernen Sie bitte im Bibliotheksmanager die Bibliothek aus dem Projekt und lassen Sie das Projekt erneut übersetzen.

4351 **Der Schrittname ist kein zulässiger Bezeichner: <Name>.**

Wählen Sie für den Schrittnamen einen gültigen Bezeichner.

4352 **Unzulässige Zeichen folgen dem zulässigen Schrittnamen: <Name>.**

Löschen Sie die unzulässigen Zeichen im Schrittnamen.

4353 **Schrittnamen sind doppelt: <Name>.**

Benennen Sie einen der Schritte um. Dieser Fehler wird auch ausgegeben, wenn ein Schritt den gleichen Namen trägt wie eine nicht-boolesche Variable.

4354 **Sprung auf nicht definierten Schritt: <Name>.**

Wählen Sie als Sprungziel einen vorhandenen Schrittnamen bzw. fügen Sie einen Schritt mit dem noch nicht definierten Namen ein.

4355 **Eine Transition darf keine Seiteneffekte (Zuweisungen, FB-Aufrufe etc.) haben.**

Eine Transition darf nur einen booleschen Ausdruck enthalten.

4356 **Sprung ohne gültige Schrittnamen: <Name>.**

Verwenden Sie einen gültigen Bezeichner als Sprungziel.

4357 **Die IEC-Bibliothek wurde nicht gefunden.**

Prüfen Sie, ob im Bibliotheksverwalter die Bibliothek `iecsfc.lib` eingebunden wurde und ob die in den Projektoptionen, Kategorie [Verzeichnisse] eingetragenen Bibliothekspfade korrekt sind.

4358 **Nicht deklarierte Aktion: <Name>.**

Sorgen Sie dafür, dass die Aktion des IEC-Schritts im Object Organizer unterhalb des AS-Bausteins eingefügt ist und der Aktionsname im Kästchen rechts vom Qualifier eingetragen ist.

4359 **Ungültiger Qualifizier: <Name>.**

Tragen Sie für die IEC-Aktion im Kästchen links neben dem Aktionsnamen einen Qualifier ein.

4360 **Erwarte Zeitkonstante nach Qualifizier: <Name>.**

Tragen Sie in das Kästchen, das sich links neben dem Namen der IEC-Aktion befindert, hinter dem Qualifier eine Zeitkonstante ein.

4361 **Bezeichner <Name> bezeichnet keine Aktion.**

Tragen Sie für die IEC-Aktion im Kästchen rechts neben dem Qualifier den Namen einer im Projekt definierten Aktion oder boolesche Variable ein.

4362 **Nicht boolescher Ausdruck in Aktion: <Name>.**

Geben Sie eine boolesche Variable oder einen gültigen Aktionsnamen ein.

4363 **IEC-Schrittname bereits für Variable verwendet: <Name>.**

Benennen Sie entweder den Schritt oder die Variable um.

4364 **Eine Transition muss ein boolescher Ausdruck sein.**

Das Ergebnis des Transitionsausdrucks muss vom Typ `BOOL` sein.

4365 **Schritt <Name> hat fehlerhaften Zeitgrenzenwert.**

Öffnen Sie den Dialog [Schrittattribute] für den Schritt <Name> und tragen Sie gültige Zeitvariablen oder Zeitkonstanten ein.

4366 **Die Marke für den Parallelschritt ist kein zulässiger Bezeichner: <Name>**

Tragen Sie neben dem Dreieck, das die Sprungmarke anzeigt, einen zulässigen Bezeichner ein.

4367 **Die Marke <Name> ist bereits vorhanden.**

Sie haben bereits eine Sprungmarke oder einen Schritt mit diesem Namen bezeichnet. Benennen Sie dementsprechend um.

4368 **Aktion <Name> wird in mehreren übereinanderliegenden SFC-Ebenen verwendet!**

Sie verwenden die Aktion `<Name>` sowohl im Baustein als auch in einer oder mehreren Aktionen dieses Bausteins.

4369 **Genau ein Netzwerk für Transitionen nötig.**

Sie haben für die Transition mehrere FUP- bzw. KOP-Netzwerke verwendet. Reduzieren Sie auf genau ein Netzwerk.

4370 **Überflüssige Zeilen nach korrekter AWL-Transition gefunden.**

Löschen Sie die nicht benötigten Zeilen am Ende der Transition.

4371 **Überflüssige Zeichen nach gültigem Ausdruck: <Name>.**

Löschen Sie die nicht benötigten Zeichen am Ende der Transition.

4372 **Schritt <Schrittname>: Zeitgrenzenwert muss vom Typ TIME sein.**

Definieren Sie die Zeitgrenzen des Schritts in den Schrittattributen mit einer Variablen vom Typ `TIME` oder mit einer Zeitangabe im korrekten Format, z. B. `t#200ms`.

4373 **IEC-Aktionen sind nur bei AS-Bausteinen erlaubt.**

Sie haben unterhalb eines Nicht-AS-Bausteins eine Aktion angelegt, die in AS programmiert ist und IEC-Aktionen enthält. Ersetzen Sie diese Aktion durch eine, die keine IEC-Aktionen enthält.

4374 **Schritt erwartet anstelle von Transition <Transitionsname>.**

Der AS-Baustein ist korrupt, eventuell aufgrund von Export-Import-Aktionen.

4375 **Transition erwartet anstelle von Schritt <Schrittname>.**

Der AS-Baustein ist korrupt, eventuell aufgrund von Export-Import-Aktionen.

4376 **Schritt erwartet nach Transition <Name>.**

Der AS-Baustein ist korrupt, eventuell aufgrund von Export-Import-Aktionen.

4377 **Transition erwartet nach Schritt <Name>.**

Der AS-Baustein ist korrupt, eventuell aufgrund von Export-Import-Aktionen.

4400 **Baustein <Name> unvollständig / mit Fehlern importiert bzw. konvertiert.**

Der Baustein kann nicht vollständig entsprechend IEC 61131-3 konvertiert werden.

4401 **S5-Zeitkonstante <Anzahl> Sekunden zu groß (max. 9990s).**

Im Akku steht keine gültige BCD-kodierte Zeit.

4402 **Direkter Zugriff nur auf E/As erlaubt.**

Stellen Sie sicher, dass Sie nur auf eine als Ein- oder Ausgang definierte Variable zugreifen.

4403 **Ungültiger oder nicht nach IEC 61131-3 konvertierbarer STEP5/7-Befehl.**

Nicht jeder STEP5/7-Befehl ist nach IEC 61131-3 konvertierbar, z. B. CPU-Befehle wie `MAS`.

4404 **Ungültiger oder nicht nach IEC 61131-3 konvertierbarer STEP5/7-Operand.**

Nicht jeder STEP5/7-Operand ist nach IEC 61131-3 konvertierbar bzw. ein Operand fehlt.

4405 **Reset eines STEP5/7-Timers kann nicht nach IEC 61131-3 konvertiert werden.**

Die entsprechenden IEC-Timer haben keinen Reset-Eingang.

4406 **STEP5/7-Zählerkonstante zu groß (max. 999).**

Im Akku steht keine gültige BCD-kodierte Zählerkonstante.

4407 **STEP5/7-Anweisung ist nicht nach IEC 61131-3 konvertierbar.**

Nicht jede STEP5/7-Anweisung ist nach IEC 61131-3 konvertierbar, z. B. `DUF`.

4408 **Bitzugriff auf Timer-/Zähler-Worte nicht IEC 61131-3 konvertierbar.**

Spezielle Timer-/Zähler-Befehle sind nicht nach IEC 61131-3 konvertierbar.

4409 **Inhalt von Akku1 oder Akku2 undefiniert, nicht nach IEC 61131-3 konvertierbar.**

Ein Befehl, der die beiden Akkus verknüpft, kann nicht konvertiert werden, weil die Akku-Inhalte nicht bekannt sind.

4410 **Aufgerufener Baustein nicht im Projekt.**

Importieren Sie zuerst den aufgerufenen Baustein.

4411 **Fehler in globaler Variablen-Liste.**

Überprüfen Sie bitte die SEQ-Datei.

4412 **Interner Fehler Nr.11.**

Wenden Sie sich bitte an Ihren Steuerungshersteller.

4413 **Fehlerhaftes Format einer Zeile in Datenbaustein.**

Im Code, der importiert werden soll, ist ein fehlerhaftes Datum enthalten.

4414 **FB/FX-Name fehlt.**

In der Ausgangs-S5D-Datei fehlt der symbolische Name eines (erweiterten) Funktionsbausteins.

4415 **Befehl nach Bausteinende nicht erlaubt.**

Ein geschützter Baustein kann nicht importiert werden.

4416 **Ungültiger Befehl.**

Der S5/S7-Befehl kann nicht disassembliert werden.

4417 **Kommentar nicht abgeschlossen.**

Schließen Sie den Kommentar mit "*)".

4418 **FB/FX-Name zu lang (max. 8 Zeichen).**

Der symbolische Name eines (erweiterten) Funktionsbausteins ist zu lang.

4419 **Erwartetes Zeilenformat ""(* Name: <FB/FX-Name> *)"".**

Korrigieren Sie die Zeile entsprechend.

4420 **FB/FX-Parametername fehlt.**

Überprüfen Sie die Funktionsbausteine.

4421 **FB/FX-Parameterartname ungültig.**

Überprüfen Sie die Funktionsbausteine.

4422 **FB/FX-Parameterart nicht angegeben.**

Überprüfen Sie die Funktionsbausteine.

4423 **Ungültiger Aktualoperand.**

Überprüfen Sie die Schnittstelle des Funktionsbausteins.

4424 **Warnung: Aufgerufener Baustein nicht vorhanden oder Kopf fehlerhaft oder hat keine Parameter**.

Der aufgerufene Funktionsbaustein wurde entweder noch nicht importiert oder ist fehlerhaft oder hat keine Parameter (in letzterem Fall können Sie die Meldung ignorieren).

4425 **Sprungmarke nicht definiert.**

Das Ziel eines Sprungs ist nicht angegeben.

4426 **Baustein hat keinen gültigen STEP5-Namen wie z. B. PB10.**

Ändern Sie den Bausteinnamen.

4427 **Timertyp nicht angegeben.**

Fügen Sie eine Deklaration des Timers in die globale Variablenliste ein.

4428 **Maximale STEP5/7-Klammertiefe überschritten.**

Es dürfen nicht mehr als sieben öffnende Klammern verwendet werden.

4429 **Fehler in Formal-Parameter-Name.**

Der Parametername darf nicht länger als vier Zeichen sein.

4430 **Typ von Formal-Parameter nicht IEC-konvertierbar.**

Timer, Zähler und Bausteine können nicht als Formal-Parameter in IEC 61131-3 konvertiert werden.

4431 **Zu viele VAR_OUTPUT-Parameter für einen Aufruf in STEP5/7-AWL.**

Ein Baustein darf nicht mehr als 16 Formal-Parameter als Ausgänge enthalten.

4432 **Sprungmarken mitten in einem Ausdruck sind verboten.**

In IEC 61131-3 dürfen Sprungmarken nicht an beliebiger Stelle stehen.

4434 **Zu viele Labels.**

Ein Baustein darf nicht mehr als 100 Labels enthalten.

4435 **Nach Sprung / Aufruf kann nicht weiterverknüpft werden.**

Nach einem Sprung oder Aufruf muss ein Ladebefehl stehen.

4436 **Inhalt von VKE undefiniert, nicht nach IEC 61131-3 konvertierbar.**

Ein Befehl, der das VKE verwendet, kann nicht konvertiert werden, weil der Wert des VKE nicht bekannt ist.

4437 **Typ von Befehl und Operand passen nicht zusammen.**

Ein Bit-Befehl wurde auf einen WORD-Operanden angewendet oder umgekehrt.

4438 **Kein Datenbaustein aufgeschlagen (fügen Sie ein A-DB ein).**

Fügen Sie einen A-Datenbaustein ein.

4500 **Unbekannte Variable oder Adresse.**

Diese Watch-Variable ist im Projekt nicht deklariert. Um die Eingabehilfe zu deklarierten Variablen aufzurufen, drücken Sie die <F2>-Taste.

4501 **Einem gültigen Watchausdruck folgen unzulässige Zeichen.**

Entfernen Sie die überzähligen Zeichen.

4520 **Fehler in Compilerdirektive: Flag erwartet vor <Name>!**

Das Pragma ist nicht korrekt eingegeben. Überprüfen Sie, ob <Name> ein gültiges Flag ist.

4521 **Fehler in Compilerdirektive: Unerwartetes Element <Name>!**

Überprüfen Sie das Pragma auf korrekte Zusammensetzung.

4522 **"flag off"-Direktive erwartet!**

Das Ausschalten des Pragmas fehlt, fügen Sie eine "flag off"-Anweisung ein.

4523 **Pragma {<Pragmaname>} in <name> nicht zulässig.**

Das Pragma kann an dieser Stelle nicht verwendet werden. Informationen zur korrekten Verwendung von Pragmas finden Sie auf Seite 212 (→ Kap. 5.3, Pragma-Anweisungen).

4550 **Index nicht im erlaubten Bereich : Variablen OD <Nummer>, Zeile <Zeilennummer>.**

Stellen Sie sicher, dass der Subindex in dem Bereich ist, der in den Zielsystemeinstellungen, Kategorie [Netzfunktionen] festgelegt ist.

4551 **Subindex nicht in erlaubten Bereich: Variablen OD <Nummer>, Zeile <Zeilennummer>.**

Stellen Sie sicher, dass der Subindex in dem Bereich ist, der in den Zielsystemeinstellungen, Kategorie [Netzfunktionen] festgelegt ist.

4552 **Index nicht in erlaubtem Bereich: Parameter OD <Nummer>, Zeile <Zeilennummer>.**

Stellen Sie sicher, dass der Subindex in dem Bereich ist, der in den Zielsystemeinstellungen, Kategorie [Netzfunktionen] festgelegt ist.

4553 **Subindex nicht in erlaubtem Bereich: Parameter OD <Nummer>, Zeile <Zeilennummer>.**

Stellen Sie sicher, dass der Subindex in dem Bereich ist, der in den Zielsystemeinstellungen, Kategorie [Netzfunktionen] festgelegt ist.

4554 **Variablenname ungültig: Variablen OD <Nummer>, Zeile <Zeilennummer>.**

Geben Sie im Feld [Variable] eine gültige Projektvariable ein. Verwenden Sie die Schreibweise `<Bausteinname>.<Variablenname>` bzw. für globale Variablen `.<Variablenname>`.

4555 **Leeres Tabellenfeld, Eingabe nicht optional: Parameter OD <Zahl>, Zeile <Zahl>.**

Für das Tabellenfeld muss eine Eingabe vorgenommen werden.

4556 **Leeres Tabellenfeld, Eingabe nicht optional: Variablen OD <Zahl>, Zeile <Zahl>.**

Für das Tabellenfeld muss eine Eingabe vorgenommen werden.

4557 **Der benötigte Parameterspeicher ist zu groß.**

Die durch das Zielsystem definierte maximale Größe der Daten, die über Parameterlisten ins Zielsystem geladen werden können, ist überschritten worden. Eine Ausgabe der Datengröße wird beim Kompilieren des Projekts im Meldungsfenster angegeben. Verringern Sie den Umfang der Parameterlisten.

4558 **Der benötigte Variablenspeicher ist zu groß.**

Die durch das Zielsystem definierte maximale Größe der Daten, die über Parameterlisten in das Zielsystem geladen werden können, ist überschritten worden. Eine Ausgabe der Datengröße wird beim Kompilieren des Projekts im Meldungsfenster angegeben. Verringern Sie den Umfang der Parameterlisten.

4560 **Ungülter Wert: Verzeichnis <Name>, Zeile <Zeilennummer>.**

Prüfen Sie den Eintrag. Welche Einträge zulässig sind, hängt von der Definition der Spalte [Attribut] in der zielsystemspezifischen XML-Beschreibungsdatei des Parameter-Managers ab bzw. von den Standardeinstellungen, die verwendet werden, wenn keine XML-Beschreibungsdatei vorliegt.

4561 **Spalte nicht definiert: <Name>.**

Einträge in einer Spalte der Parameterliste beziehen sich auf die hier genannte Spalte, die jedoch nicht definiert ist. Die Spaltendefinitionen sind in der XML-Beschreibungsdatei des Parameter-Managers für das aktuelle Zielsystem enthalten. Wenn eine solche XML-Beschreibungsdatei nicht vorhanden ist, gelten die Standardeinstellungen.

4562 **Der Index/Subindex <Name> wird bereits verwendet: Verzeichnis <Name>, Zeile <Zeilennummer>.**

Eine Index/Subindex-Paarung muss innerhalb aller Parameterlisten eindeutig sein, da sie für den Zugriff verwendet werden kann. Ändern Sie die Indizierung entsprechend.

4563 **Der Name <Name> wird bereits verwendet: Verzeichnis <Name>, Zeile <Zeilennummer>.**

Der Name für einen Eintrag muss innerhalb aller Parameterlisten eindeutig sein, da er für den Zugriff verwendet werden kann. Verwenden Sie einen anderen Namen.

4564 **Index <Name> nicht in erlaubtem Bereich: Verzeichnis <Name>, Zeile <Zeilennummer>.**

Geben Sie in dem Feld einen Index an, der in dem Bereich liegt, der in den Zielsystemeinstellungen, Kategorie [Netzwerkfunktionen] im Feld [Indexbereich] für den vorliegenden Listentyp (Variablen, Parameter, Mappings) definiert ist.

4565 **Subindex <Name> nicht in erlaubtem Bereich: Verzeichnis <Name>, Zeile <Zeilennummer>.**

Geben Sie in dem Feld einen Subindex an, der in dem Bereich liegt, der in den Zielsystemeinstellungen, Kategorie [Netzwerkfunktionen] im Feld [Subindexbereich] definiert ist.

4566 **Fehler beim Importieren des Parameter-Managers.**

Sie haben eine Exportdatei ins Projekt importiert, die fehlerhafte Information zum Parameter Manager enthält. Überprüfen Sie die `*.exp`-Datei dahingehend.

4600 **Netzwerkvariablen: <Name> ist kein boolescher Ausdruck!**

Stellen Sie sicher, dass die Variable, die im Dialog [Eigenschaften] einer Netzwerkvariablenliste bei der Option [Ereignisgesteuerte Übertragung] angegeben wurde, vom Typ `BOOL` ist.

4601 **Netzwerkvariablen "<CAN|UDP>": Es ist keine zyklische oder freilaufende Task zum Netzwerkvariablenaustausch vorhanden.**

Es gibt keine zyklische oder freilaufende Task bzw. `PLC_PRG` im Projekt, in dem die Netzwerkvariablen der genannten Liste vom Typ `CAN` bzw. `UDP` verwendet werden. Eine Deklaration allein ist nicht ausreichend!

Sorgen Sie dafür, dass die Variablen in einer entsprechenden Task bzw. in `PLC_PRG` verwendet werden. Falls Sie die Variablen in mehreren Tasks verwenden, wird beim Variablenaustausch die Task mit der höchsten Priorität berücksichtigt.

4602 **<Name der Netzwerkvariablenliste>: Das Objekt verwendet UDP-Port <Portnummer> und nicht <Portnummer>.**

In den Einstellungen der genannten Netzwerkvariablenliste wird eine Portnummer verwendet, die nicht mit der ersten Portnummer übereinstimmt, die in allen vorhandenen globalen NW-Variablenlisten gefundenen wurde. Sorgen Sie dafür, dass alle Netzwerkvariablenlisten die gleiche Portnummer verwenden.

4620 Im Projekt wurden unbenutzte Variablen gefunden. Lesen Sie hierzu die Beschreibung auf Seite 143 (→ Abschnitt. Unbenutzte Variablen).

4621 Bei der Zuweisung von Variablen auf bestimmte Speicherbereiche über `AT`-Deklaration wurden Überlappungen festgestellt. Lesen Sie hierzu die Beschreibung auf Seite 143 (→ Abschnitt. Überlappende Speicherbereiche).

4622 In mehr als einer Task werden IEC-Adressen referenziert, die auf den gleichen Speicherbereich weisen. Lesen Sie hierzu die Beschreibung ab Seite 143 (→ Abschnitt. Konkurrierender Zugriff).

4623 Im Projekt wird auf denselben Speicherbereich an mehr als einer Stelle schreibend zugegriffen. Lesen Sie hierzu die Beschreibung auf Seite 144 (→ Abschnitt. Mehrfaches Speichern auf Output).

4650 **AxisGroup <Name>: Task <Name> existiert nicht.**

In der Steuerungskonfiguration ist in der Definition der Achsgruppe (Dialog [Module parameters], Spalte [Value]) für die Task, die den Datentransfer der Achsgruppe steuert, ein Name angegeben, der in der Taskkonfiguration nicht bekannt ist. Korrigieren Sie die Task- bzw. Steuerungskonfiguration entsprechend.

4651 **AxisGroup <Name>: Zykluszeit (dwCycle) nicht eingestellt.**

Tragen Sie im Dialog [Module parameters] der Achsgruppe in der Steuerungskonfiguration einen Wert für die Zykluszeit (`dwCycle`) ein.

4652 **Drive <Name>: wDriveID in dieser Achsgruppe schon vorhanden.**

In der Steuerungskonfiguration liegt innerhalb der Achsgruppe bereits ein Antrieb mit gleicher Drive-ID vor. Markieren Sie den Antriebseintrag im Konfigurationsbaum und vergeben Sie im Dialog [Drive] eine innerhalb der Achsgruppe eindeutige Drive-ID.

4670 **CNC-Programm <Name>: Globale Variable <Name> nicht gefunden.**

Im CNC-Programm wird eine globale Variable verwendet (z. B. `$glob_var$`), die im Projekt nicht definiert ist. Deklarieren Sie die Variable bzw. korrigieren Sie die Variablenzuweisung im CNC-Programm.

4671 **CNC-Programm <Name>: Variable <Name> hat falschen Typ.**

Sie verwenden im CNC-Programm bei einem Fahrbefehl eine Variable, die mit einem an dieser Stelle nicht zulässigen Typ deklariert ist. Korrigieren Sie die Verwendung bzw. Typdeklaration.

4685 **Kurvenscheibe <Name>: Stützpunkttabellen-Datentyp unbekannt.**

Prüfen Sie im CAM-Editor im Dialog [Übersetzungsoptionen] den Datentyp für die äquidistante oder elementoptimierte Stützpunkttabelle.

4686 **Kurvenscheibe <Name>: Stützpunkt überschreitet Datentyp-Bereich.**

In der Kuvenscheibe werden Stützpunkte verwendet, die nicht mehr im Datenbereich liegen, der für die Stützpunkttabelle definiert ist. Lesen Sie hierzu die Definition im Dialog [Übersetzungsoptionen] im CAM-Editor.

4700 **<Nummer> (<Name>): Watchausdruck <Name> ist keine Nummer.**

Sie verwenden in der Konfiguration der Visualisierung eine Variable, die keine Zahl definiert, obwohl es an dieser Stelle gefordert ist (z. B. bei Konfiguration von XOffset oder Winkel).

4701 **<Name> (<Zahl>): Watchausdruck <Name> ist nicht vom Typ BOOL.**

Sie verwenden in der Konfiguration der Visualisierung eine Variable, die nicht vom Typ `BOOL` ist, obwohl es an dieser Stelle gefordert ist.

4702 **<Name> (<Zahl>): Watchausdruck <Name> ist nicht vom Typ STRING.**

Sie verwenden in der Konfiguration der Visualisierung eine Variable, die nicht vom Typ `STRING` ist, obwohl es an dieser Stelle gefordert ist.

4703 **<Name> (<Nummer>): Ungültiger Watchausdruck <Name>.**

Die Visualisierung enthält eine ungültige Variable.

4704 **<Name> (<Zahl>): Fehlerhafter Initialwert innerhalb der Watchliste <Name>.**

In einer visualisierten Watchliste (`INTERN`-Befehl in Kategorie [Eingabe]) ist ein fehlerhafter Initialwert enthalten. Prüfen Sie die verwendete Liste.

4900 **Unzulässiger Typ für Konvertierung.**

Die verwendete Typkonvertierung wird nicht eingestellten Codegenerator nicht unterstützt wird.

4901 **Interner Fehler: Überlauf in Array access!**

Die Array-Grenzen sind zu groß für eine 32-Bit-Variable. Verkleinern Sie den Array-Index-Bereich.

5100 **<Bausteinname> (<Zeilennummer>): Ausdruck zu komplex. Keine Register mehr verfügbar.**

Der genannte Ausdruck ist so komplex, dass die verfügbaren Register nicht ausreichen. Versuchen Sie, den Ausdruck durch Verwendung von Zwischenvariablen zu vereinfachen.

12 Index

A

Ablaufsprache 33
- *Aktion* 33
  - *Ausgangsaktion* 34
  - *Eingangsaktion* 34
- *Alternativzweig* 39
- *Flags* 38
- *IEC-Schritt* 35, 37
- *Parallelzweig* 39
- *Schritt* 33
- *Schritt, aktiver* 35
- *Transition* 34
- *Transitionsbedingung* 34
- *Variablen, implizite* 37

Aktion 22
- *Ablaufsprache* 33
- *AS* 33
- *Aufruf* 22
- *Ausgangsaktion* 34
- *Eingangsaktion* 34

Aktion hinzufügen 163

Alternativzweig
- *Ablaufsprache* 39
- *AS* 39

Anweisungen
- *ST* 27
- *Strukturierter Text* 27

Anweisungsliste 24
- *Identifikator-Marke* 24
- *Klammern* 26
- *Kommentare* 24
- *Modifikatoren* 25
- *Operatoren* 25
- *Whitespace* 24

Arbeitsbereich 71

Arbeitsgruppen 144

Archiv 103
- *erstellen* 105

Array 392

AS 33
- *Aktion* 33
  - *Ausgangsaktion* 34
  - *Eingangsaktion* 34
- *Alternativzweig* 39
- *Flags* 38
- *IEC-Schritt* 35, 37
- *Parallelzweig* 39
- *Schritt* 33
- *Schritt, aktiver* 35
- *Transition* 34
- *Transitionsbedingung* 34
- *Variablen, implizite* 37

Aufrufbaum ausgeben 166

Aufzählungstyp, Enumeration 395

Ausdrücke
- *ST* 26, 27
- *Strukturierter Text* 26, 27

Ausloggen 180

AWL 24
- *Klammern* 26
- *Kommentare* 24
- *Modifikatoren* 25
- *Operatoren* 25
- *Whitespace* 24

B

Bibliothek "standard.lib" 402

Bibliotheken 23

Bibliotheken einbinden 51

Bildschirmteiler 70

Bistabile Funktionsblöcke 408
- *RS* 409
- *SEMA* 410
- *SR* 408

Breakpoint 43
- *Dialog* 183

Breakpoint ein/aus 182

C

CFC 40

CheckBounds 393

CheckPointer 394

CheckRangeSigned 398

CheckRangeUnsigned 398

CONCAT 402

CTD 414

CTU 413

CTUD 415

D

DATE 391

DATE_AND_TIME 392

Datentyp einfügen 159

Datentypen 23
- *Array* 392
- *Aufzählungstyp, Enumeration* 395
- *Pointer* 394
- *REAL und LREAL* 390
- *Referenzen* 397
- *STRING* 390
- *Strukturen* 396
- *Unterbereichstypen* 397
- *Zeitdatentypen* 390

Debugging 13, 43
- *Breakpoint* 43
- *Einzelschritt* 43
- *Einzelzyklus* 43
- *Logbuch* 45
- *Monitoring* 43
- *Online* 43
- *Simulation* 45

Vorausetzungen 43
DELETE 403
Drehzahlregelung programmieren 60

E

Editierfunktionen, allgemeine 167
Ein-/Ausschalten des Antriebsumrichters 54
Eingabehilfe 173
Einloggen 178
Einstellungen Dokumentation 107
Objekt einfügen 108
Ersetzen 142
EXIT
ST 32
Strukturierter Text 32

F

FIND 403
Flags
Ablaufsprache 38
AS 38
Flankenerkennung 411
R_TRIG 411, 412
FOR
ST 30
Strukturierter Text 30
Freigrafischer Funktionsplaneditor 40
Funktion
CONCAT 402
DELETE 403
FIND 403
INSERT 404
LEFT 405
LEN 405
MID 406
REPLACE 406
RIGHT 407
Funktionen
CheckBounds 393
CheckPointer 394
CheckRangeSigned, CheckRangeUnsigned 398
Funktionsblock
Aufruf 17
ST 28
Strukturierter Text 28
Instanz 16
Kontaktplan 42
KOP 42
Funktionsleiste 70
Funktionsplan 40
FUP 40

G

Ganzzahlige Datentypen 389
Globale Variablenliste einfügen 159
Globales Ersetzen 142
Globales Suchen 141

H

Hinweise, wichtige 11

I

IEC 61131-3 45
IEC-Schritt
Ablaufsprache 35, 37
AS 35, 37
IF
ST 29
Strukturierter Text 29
INSERT 404
Instanz öffnen 164

K

Kennworte 88
Klammern
Anweisungsliste 26
AWL 26
Kommentare
Anweisungsliste 24
AWL 24
Kommunikation aufbauen 52
Kontakt
Kontaktplan 41
KOP 41
Kontaktplan 41
Funktionsblock 42
Kontakt 41
Programm 42
Spule 41
Reset 42
Set 42
Kontextmenü 72
KOP 41
Funktionsblock 42
Kontakt 41
Programm 42
Spule 41
Reset 42
Set 42
KOP als FUP 42

L

LEFT 405
LEN 405
Logbuch 45, 83
LREAL 390

M

Makrobibliotheken 98
Makros 97
bearbeiten 98
definieren 97
löschen 98
Makrobibliotheken 98
MC_ConnectAxis_PD 421

Meldungsfenster 71
Menüleiste 69
MID 406
Modifikatoren
Anweisungsliste 25
AWL 25
Monitoring 43
Motorachse positionieren 55

O

Object Organizer 70
Objekt
bearbeiten 161
Eigenschaften 161
konvertieren 160
kopieren 160
umbenennen 159
Objekte verwalten 156
Online Change 179
Operatoren
Anweisungsliste 25
AWL 25

P

Parallelzweig
Ablaufsprache 39
AS 39
Passwörter 88
PLC_PRG 21
PLC-Editor
Elemente 13
PLC-Editor Bibliothek "MPLCProcessdata.lib" 421
PLC-Editor Datentypen 389
Pointer 394
Programm
Aufruf
Parameterwerte 20
Kontaktplan 42
KOP 42
PLC_PRG 21
Programm testen 56, 61, 66
Programmierbeispiel
Drehzahlregelung einer Motorachse mit Geber 59
Positionieren einer Motorachse 47
Tippbetrieb einer Motorachse ohne Geber 64
Programmierbeispiele 46
Programmiersprachen 24
Ablaufsprache 33
Anweisungsliste 24
AS 33
AWL 24
CFC 40
Freigrafischer Funktionsplaneditor 40
Funktionsplan 40
FUP 40
Kontaktplan 41
KOP 41
ST 26
Strukturierter Text 26
unterstützte 24
Projekt
Abarbeitung
Breakpoint ein/aus 182
Reset (kalt) 182
Reset (Ursprung) 182
Werte zurücksetzen 181
Abarbeitung starten 181
Abarbeitung stoppen 181
alles bereinigen 124
alles übersetzen 124
Arbeitsgruppen 144
Archiv 103
erstellen 105
Bestandteile 13
Baustein 13
Funktion 14
Funktionsblock 16
Programm 19
Breakpoint
Dialog 183
dokumentieren 130
Download-Information laden 124
drucken 106
Einstellungen Dokumentation 107
Objekt einfügen 108
erstellen 12
exportieren 131
importieren 132
in andere Sprache übersetzen 125
in Steuerung laden 180
kopieren 139
neu aus Vorlage 99
neues Projekt 99
öffnen 99
öffnen aus der Projektdatenbank 101
öffnen aus der Steuerung 100
Projektdatenbank 145
alles abrufen 150
festlegen 147
gemeinsame Objekte einfügen 153
Login 154
mehrfach auschecken 150
mehrfach auschecken rückgängig 151
mehrfach einchecken 151
mehrfach festlegen 150
Objekt abrufen 148
Objekt auschecken 148
Objekt auschecken rückgängig 148
Objekt einchecken 148
Objekt Unterschiede 148
Objekt Versionsgeschichte 149
Projekt Version labeln 153
Projekt Versionsgeschichte 151
Status auffrischen 154
Projektinformation 139
schließen 101
Siemens-Import 133
speichern 101
speichern unter 102

Struktur 12
testen 12
überprüfen 143
übersetzen 122
übersetzt darstellen 128
Übersetzung umschalten 129
vergleichen 135, 137, 138
zuletzt geöffnete 101
Projekt erstellen 48
Projekt übertragen 56
Projektdatenbank 93, 145
alles abrufen 150
ENI konfigurieren 94
festlegen 147
gemeinsame Objekte einfügen 153
Login 154
mehrfach auschecken 150
mehrfach auschecken rückgängig 151
mehrfach einchecken 151
mehrfach festlegen 150
Objekt abrufen 148
Objekt auschecken 148
Objekt auschecken rückgängig 148
Objekt einchecken 148
Objekt Unterschiede 148
Objekt Versionsgeschichte 149
Projekt Version labeln 153
Projekt Versionsgeschichte 151
Status auffrischen 154
Projekte
verwalten 99
Projektoptionen 73
Arbeitsbereich 78
Benutzerinformation 75
Editor 76
Farben 80
Kennworte 88
Laden & Speichern 74
Logbuch 83
Makros 97
bearbeiten 98
definieren 97
löschen 98
Makrobibliotheken 98
Projektdatenbank 93
ENI konfigurieren 94
Sourcedownload 89
Symbolkonfiguration 90
Übersetzungsoptionen 85
Verzeichnisse 81

Q

Querverweisliste ausgeben 165

R

R_TRIG 411, 412
REAL 390
Referenzen 397
REPEAT
ST 31
Strukturierter Text 31
REPLACE 406
Ressourcen 21
RETURN
ST 28
Strukturierter Text 28
RIGHT 407
RS 409
RTC 420

S

Schritt
Ablaufsprache 33
AS 33
Schritt, aktiver
Ablaufsprache 35
AS 35
SEMA 410
Sicherheitshinweise 11
Simulation 45
Sourcedownload 89
Spule
Kontaktplan 41
KOP 41
Reset 42
Set 42
SR 408
ST 26
Anweisungen 27
Ausdrücke 26
Auswertung 27
EXIT 32
FOR 30
Funktionsblock
Aufruf 28
IF 29
REPEAT 31
RETURN 28
WHILE 30
Zuweisungsoperator 28
Standard-Datentypen 389
BOOL 389
Statusleiste 71
Steuerungskonfiguration einstellen 49, 59, 64
STRING 390
String Funktionen 402
Strukturen 396
Strukturierter Text 26
Anweisungen 27
Ausdrücke 26
Auswertung 27
EXIT 32
FOR 30
Funktionsblock
Aufruf 28
IF 29
REPEAT 31

RETURN 28
WHILE 30
Zuweisungsoperator 28
Suchen 141
Symbolkonfiguration 90

T

TIME 391
TIME_OF_DAY 391
Timer 417
RTC 420
TOF 419
TON 418
TP 417
Tippbetrieb programmieren 65
TOF 419
TON 418
TP 417
Traceaufzeichnung 57, 62, 67
Transition
Ablaufsprache 34
AS 34
Transitionsbedingung
Ablaufsprache 34
AS 34

U

Unterbereichstypen 397

V

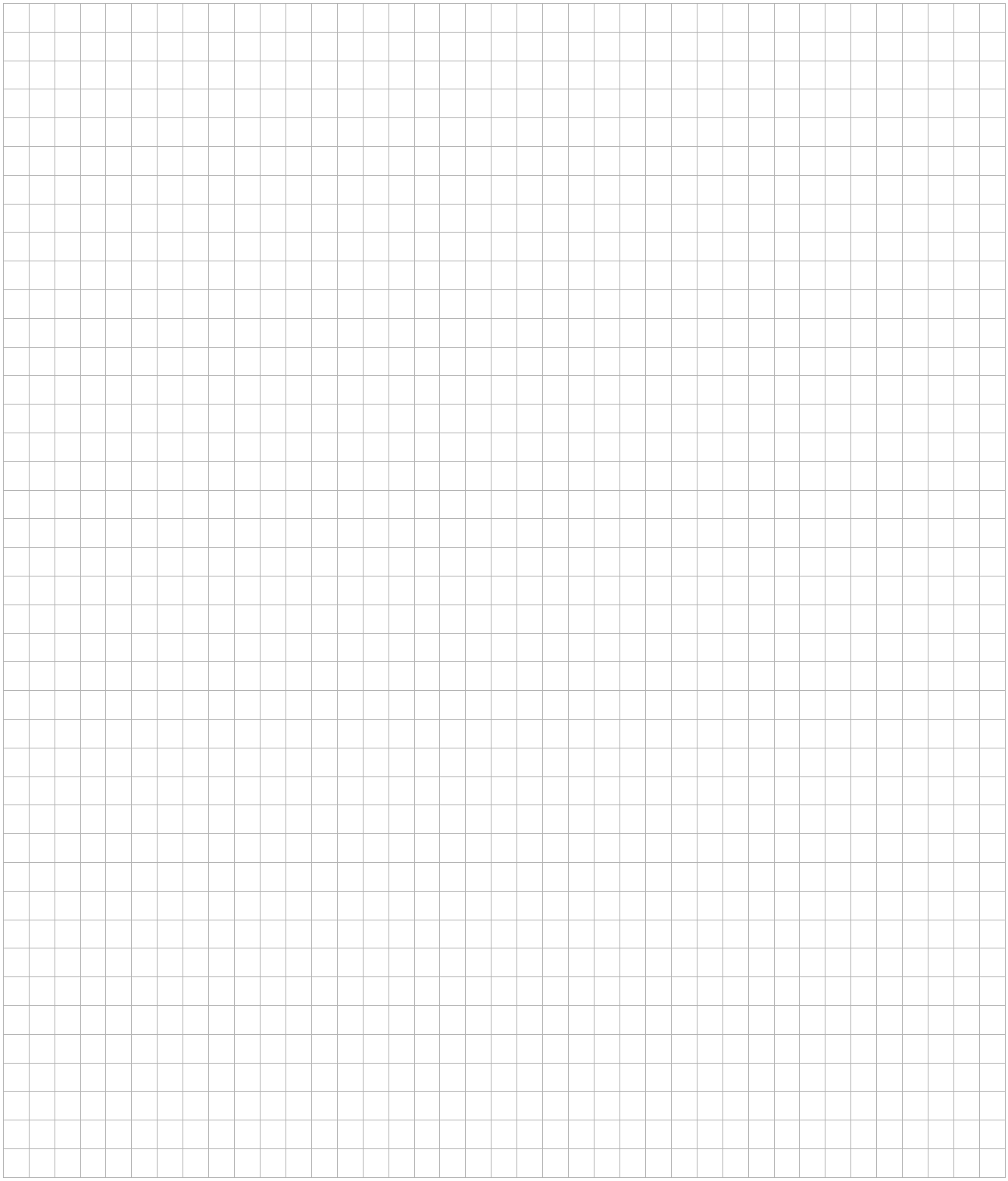
Variablen, implizite
Ablaufsprache 37
AS 37
Variablendeklaration 176
Voraussetzungen für Programmierbeispiele 46

W

Warnhinweise 11
Watch- und Rezepturverwalter, Tracekonfiguration 175
WHILE
ST 30
Strukturierter Text 30
Whitespace
Anweisungsliste 24
AWL 24

Z

Zähler 413
CTD 414
CTU 413
CTUD 415
Zeitdatentypen 390
DATE 391
DATE_AND_TIME 392
TIME 391
TIME_OF_DAY 391
Zuweisungsoperator
ST 28
Strukturierter Text 28



SEW EURODRIVE

Wie man die Welt bewegt

Mit Menschen, die schneller richtig denken und mit Ihnen gemeinsam die Zukunft entwickeln.

Mit einem Service, der auf der ganzen Welt zum Greifen nahe ist.

Mit Antrieben und Steuerungen, die Ihre Arbeitsleistung automatisch verbessern.

Mit einem umfassenden Know-how in den wichtigsten Branchen unserer Zeit.

Mit kompromissloser Qualität, deren hohe Standards die tägliche Arbeit ein Stück einfacher machen.

SEW-EURODRIVE
Driving the world

Mit einer globalen Präsenz für schnelle und überzeugende Lösungen. An jedem Ort.

Mit innovativen Ideen, in denen morgen schon die Lösung für übermorgen steckt.

Mit einem Auftritt im Internet, der 24 Stunden Zugang zu Informationen und Software-Updates bietet.

SEW EURODRIVE

SEW-EURODRIVE GmbH & Co KG
P.O. Box 3023 · D-76642 Bruchsal / Germany
Phone +49 7251 75-0 · Fax +49 7251 75-1970
sew@sew-eurodrive.com

→ **www.sew-eurodrive.com**